General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

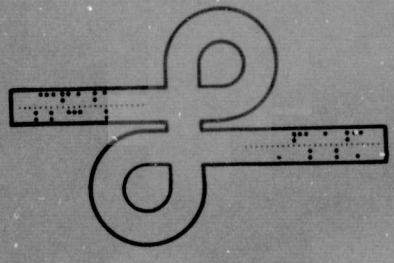
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NASA CR-
147393



LINKABIT CORPORATION

10453 ROSELLE STREET
UNIVERSITY INDUSTRIAL PARK
SAN DIEGO, CALIFORNIA 92121

LINKABIT CORPORATION
10453 Roselle Street
San Diego, CA 92121

FINAL REPORT

for the

DIGITAL TV PROCESSING SYSTEM

26 November 1975

Contract NAS9-14561
DRL Item No. 2

Submitted to:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lyndon B. Johnson Space Center
Houston, Texas 77058

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF FIGURES

TABLE OF TABLES

ABSTRACT

Two digital video data compression systems are provided in this report which are directly applicable to the Space Shuttle TV Communication System.

1) For the uplink a low rate, 1 Megabit/sec, monochrome video data compressor is used. The data compression is achieved by using a motion detection technique in the Hadamard domain. The resultant source rate is directly proportional to the motion contents of the reproducing scene. To transform the variable source rate into a fixed rate, an adaptive rate buffer is provided. The rate buffer size is approximately 600K bits and is required only at the encoder.

2) For the downlink a color video data compressor is considered. The data compression is achieved first by intra-color transformation of the original signal vector, (R(t), G(t), B(t)), into a vector, (Y(t), I(t), Q(t)), which has lower information entropy. Then two-dimensional data compression techniques are applied to the Hadamard transformed components of Y(t), I(t) and Q(t). The resultant source rate is a fixed rate of approximately 24 Megabits/sec.

Mathematical models and data reliability analyses are also provided for the above video data compression techniques transmitted over a channel encoded Gaussian

channel. It is shown that substantial gains can be achieved by the combination of video source and channel coding.

1.0 Introduction

This report presents the results of a study on digital TV processing suitable for Space Shuttle communication purposes. This TV communication system is divided into two parts: uplink and downlink transmissions. A monochrome video source coding utilizing two-dimensional Hadamard transformation and motion detection methods is used for the uplink TV transmission, where the data rate is very low. A color video source coding technique utilizing a two-dimensional Hadamard transform is used for the downlink TV transmission. The source data rates are 1 megabit per second for the uplink and approximately 24 megabits per second for the downlink.

1.1 Background

In the past, due to hardware considerations, television communication for space flight systems has resorted exclusively to analog processing and transmission techniques. With the advances in digital communication techniques, in particular the recent low dissipation mass memory technologies, digital television transmission offers several potential advantages over analog transmission, such as, increase in data link efficiency, higher picture reliability, less hardware complexity, elimination of bulky transmitting antenas, ease of multiplexing with other digital data, etc. An efficient video source coding technique used in conjunction with an

efficient channel encoding technique can provide significant performance advantages over the analog transmission system. It should be emphasized that no universally optimum video data compression schemes exist for arbitrary systems. Generally speaking, a higher data compression ratio requires more complex hardware. Thus, an optimal choice of video source encoder-decoder often depends on the environment of the system under consideration. Some of the constraints and obvious hardware considerations for Space Shuttle TV communications are given in the following paragraphs.

1.2 Basic System Constraints and Hardware Considerations

The purpose of the Space Shuttle TV communications is to provide bi-directional transmission of images between a ground station and a space vehicle. The data transmitted is linked via the TDRS satellite. The "uplink" transmission is defined as the transmission from a ground station to a space vehicle via the TDRS. The "downlink" transmission is likewise defined as the data transmission from a space vehicle to a ground station via the TDRS. The basic constraints for the source data rates are given as follows:

1) Uplink Transmission: 1 Megabit per second.

2) Downlink Transmission: 50 Megabits per second.

The TV signals transmitted during the uplink mode consists primarily of scenes that are highly stationary. Due to the extremely low transmission data rate and low

rate of object displacements, the major recommendation of this study is to implement a motion detection algorithm that utilizes the advantages of data redundancies resulting from the stationary objects. Only monochrome TV signals are transmitted in this mode.

The TV signal transmitted in the downlink mode consists of high quality video information gathered by the space vehicle. Color, fine picture detail, and motion quality are important considerations in this case.

In the design of the Space Shuttle TV communications system, it is important to differentiate the hardware environments where the encoders and decoders are located. For the space vehicle, where one end of the communications links is located, is obviously limited in terms of hardware wieght, power comsumption and hardware size, whereas in the ground station, these constraints can be greatly relaxed. Thus, in this report, special considerations are given to the design of the video source encoder for the downlink transmission, and the video source decoder for the uplink transmission.

1.3 Scope of this Report

This report will provide viable video data compression schemes for both the uplink and the downlink TV transmissions. This study is basically divided into three parts: (1) two-dimensional monochrome video data compression, (2) mono-

chrome video data compression using motion detection method, and (3) color video data compression method. The two-dimensional monochrome data compression method has been studied and simulated in previous LINKABIT work entitled "Study of Efficient Video Compression Algorithm for Space Shuttle Applications," Final Report [1]. The monochrome video data compression using motion detection method has been computer simulated on the LIM Video Controller for subjective quality observations.

2.0 Monochrome Video Data Compression

The art of video data compression is based exclusively upon the following two facts:

FACT 1. Among all possible two-dimensional still pictures only relatively few can be classified as recognizable to the "average" observers. The redundancy of the set of recognizable pictures mainly occurs as statistical correlations of video signals between adjacent sampling points. Other redundancies may occur in the form of psychovisual effects. For example, consider two frames of video signals, each of which is a reproduction of the same scene at different light illuminations. From the signal point of view, these two frames may be quite distinct, yet from the "constant brightness" phenomenon we know, from the vision point of view, these two frames are identical in detail with only contrast and brightness differences. Psychovisual phenomena produce the type of redundancies that two or more distinct frames of video signals may be interpreted by the average viewer as identical. These redundancies are deterministic in the sense that they can be applied whenever the situation arises. This should be distinguished from the statistical redundancy, used most often in two-dimensional video data compression schemes, obtained empirically by classifying a picture as recognizable if a number of selected viewers agree so. From video data compression point of view, these

statistics together with the psychovisual phenomena should be utilized to obtain a set of recognizable pictures in the sense of least information entropy. We shall refer to the resultant redundancy of this set as the spatial statistical redundancy of the "set of recognizable pictures." Other useful psychovisual phenomena are:

a) The logarithmic response of the visual system,

b) The human eyes are insensitive to continuous small variations of greylevels, but are greatly sensitive to abrupt changes of greylevels; this sensitiveness occurs in the form of awareness only, as opposed to the estimation of exact greylevel changes.

c) The phenomenon of "tritanopia," also known as "two-color vision," by which the eyes tend to confuse, for small colored regions, the color of purple, bluish grey, and greenish yellow, yet exhibit good accuity for orange red and cyan.

d) At normal observation distance, the spatial response of the eyes decreases as the spatial frequency, (i.e., a pattern of alternate light and dark levels), increases.

FACT 2. Similarly, among the set of possible sequences of recognizable pictures, only relatively few can be classified as comprehensible to the "average" observers. Picture sequences containing fast object movements, unrelated contents, irratic change of intensity, and many others are examples of incomprehensibility. Thus, among the set of recognizable

picture sequences, redundancy occurs not only two-dimensionally as spatial statistical redundancy, but also in the form of time statistical redundancy.

The statistical redundancies mentioned above enable a video data compressor to encode only the information pertinent to the set of recognizable picture sequences. Accordingly, an efficient video data compressor is an "intelligent" machine that is capable of identifying every psychovisual phenomenon. Such a machine must inevitably be complex in structure and generally not suitable for a limited environment such as a space flight application. The objective of this study is to provide a feasible scheme that is simple, in terms of hardware implementation, and to provide reasonable data compression ratio.

We shall begin with a study of monochrome TV signal data compression. It is directly applicable to the uplink TV communication system of the Space Shuttle program. Since color information can be most efficiently transmitted by transforming the three base color components into the monochrome brightness and chrominance plane, the monochrome technique will be equally essential to the color TV data compression. The monochrome TV data compression is basically divided into two major steps: two-dimensional data compression and motion detection. The former utilizes solely

the spatial statistical redundancy of the set of recognizable pictures. Additional properties due to the time statistical redundancy are incorporated in the motion detection technique.

2.1 Two-Dimensional Monochrome Video Data Compression

Among the many monochrome video data compression techniques, the most feasible one was originally investigated by Landau and Slepian [2]. It uses exclusively the property of spatial statistical correlation between video signals of adjacent elements. Time statistical correlation and other two-dimensional psychovisual phenomena are not invoked; consequently the resultant achievable compression ratio is not very high.. Due to its importance to the technique proposed by this study, a refined version is given in the "Study of Efficient Video Compression Algorithms for Shuttle Applications," Final Report [1], which is summarized as follows:

We may consider each frame of video signal as being sampled by an A-to-D converter. The A-to-D converter samples uniformly across each horizontal line. The sampling rate of the A-to-D converter operates at a frequency above the Nyquist frequency of the desired video bandwidth. In addition, each sampling point is linearly approximated by a sufficient number of information bits so that no degradation due to false image contouring is visible. Using 512 samples per horizontal line and 8 information bits per sample

seems sufficient for this purpose. Since each frame contains 480 visible horizontal lines (240 lines per field), we can represent each frame of the digitized video signal as a lattice of 480 x 512 sampling points each of which has integer representation between 0 and 255. Using this representation, there are $(256)^{480x512}$ possible pictures. Among these, most are not recognizable to an average observer. Considering the subset of those which are recognizable to an average observer, one of the most obvious properties of this subset is the statistical correlations between adjacent lattice points. It is clear that the statistical correlations between two different sampling points decreases as their geometrical distance increases. One method for using the advantage of this spatial statistical correlation is to partition the lattice into an aggregate of subpictures, where in each subpicture the statistical correlations between any two sampling points are high. To minimize the computational complexity, the selection of the shape of the subpictures must be uniform. One of the intuitive choices is the shape of a rectangle. Let the subpicture be a rectangle of size m x n (m vertically and n horizontally), where m and n are divisors of 480 and 512, respectively. (Note, from an equidistance (between boundaries) point of view, a hexagonal mosaic-like pattern should be optimal; however, this will induce unwanted complications in the frame edges.) Each

subpicture can be considered as an m x n dimensional random vector:

$$\underline{Y} = \begin{pmatrix} X_{11}, X_{12} \cdots X_{1n} \\ X_{21}, X_{22} \cdots X_{2n} \\ \vdots \\ X_{m1}, X_{m2} \cdots X_{mn} \end{pmatrix} \qquad (2.1)$$

One can form the ensemble of recognizable subpictures as the aggregate of those members extracted from the set of recognizable pictures. Data compression, due to the statistical redundancy of this set, can be achieved by statistical analysis. In the least mean square error sense, this corresponds to the Karhunen-Loeve procedure, which involves the construction of an orthonormal basis that diagonalizes the covariance matrix of (2.1):

$$E[Y \cdot Y^T] = \int_\Omega Y \cdot Y^T \, P_Y(\omega) d\omega \quad (mn \times mn \text{ matrix}) \qquad (2.2)$$

where Ω is the sample space consisting of 256^{mn} elements, and $P_Y(\omega)$ is the joint probability density function of Y. Bit rate reduction is then obtained by quantizing or discarding data components, in the transformed coordinates, that have low variances. Since visual sensation, where the ultimate judgement of the reproduced picture lies, is quite different

from the expected mean square error criterion, other types of orthonormal transformations were sought. In particular, Hadamard transformation was judged to be superior. In a Hadamard transformation, the basis vectors are obtained from the row vectors of a Hadamard matrix. An n x n integral matrix, H, is Hadamard of order n, if

$$H^T H = nI, \tag{2.3}$$

where H^T is the transpose of H, and I is the n x n identity matrix. Hadamard matrices only occur for orders n which satisfy the condition:

$$n = 1,\ 2 \text{ or } n \equiv 0 \bmod (4).$$

In particular, the Hadamard matrix of order 2^k can be obtained easily by the k-th tensor product of H_2:

$$H_{2^K} = \underbrace{H_2 \otimes H_2 \otimes \cdots \otimes H_2}_{k \text{ terms}} \tag{2.4}$$

where $\otimes$ is the standard tensor product notation, and

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.5}$$

Hence, when $m = 2^{K_1}$ and $n = 2^{K_2}$, the basis vectors of the Hadamard transformation can be obtained from the row vectors of $H_{2^{K_1+K_2}}$; which in turn is obtained as the $K_1 + K_2$ tensor product of H_2. The Hadamard matrix obtained this way has entries +1 or -1; therefore, the orthonormal basis can be obtained by dividing each row vector by the constant

$$\sqrt{mn} = \sqrt{2^{K_1+K_2}}\ .$$

Let us denote the basis vectors thus obtained by

$$\hat{b}_{ij} \qquad \begin{matrix} i=1, \ldots, m \\ j=1, \ldots, n. \end{matrix}$$

In particular, the first basis vector has the form:

$$\hat{b}_{11} = \frac{1}{\sqrt{mn}} \begin{pmatrix} 1 & 1 \ldots 1 \\ 1 & 1 \ldots 1 \\ \vdots & \\ 1 & \ldots \ldots 1 \end{pmatrix}$$

m x n vector

The remaining basis vectors have the form of $1/\sqrt{mn}$ multiplied by an m x n vector, half of whose entries have value +1 and the remaining half have value -1. The components of Y in the Hadamard coordinate can be obtained as follows:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} \hat{b}_{ij} \tag{2.6}$$

where

$$C_{ij} = Y \cdot \hat{b}_{ij} \tag{2.7}$$

In particular,

$$C_{11} = \frac{1}{\sqrt{mn}} \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} \tag{2.8}$$

The remaining C_{ij}'s have the form of $1/\sqrt{mn}$ multiplied by the difference of two terms, each of which is the sum of half of the X_{ij}'s. C_{11} is also referred to as the d.c. component of the subpicture for the reason that it is a

constant multiple of the sum of video signals of the sample points within the subpicture.

Conversely, given a vector, Z, in the Hadamard basis, the corresponding subpicture Y′ can be retrieved by the inverse transformation:

$$Y' = \begin{pmatrix} x'_{11}, x'_{12}, \cdots x'_{1n} \\ \vdots \\ x'_{m1}, x'_{m2}, \cdots x'_{mn} \end{pmatrix} \tag{2.9}$$

Where

$$x'_{ij} = Z \cdot \hat{b}_{ij}.$$

Equivalently, the Hadamard transformation can be considered as

$$Z = \begin{pmatrix} C_{11}, \cdots C_{1n} \\ \vdots \\ C_{m1}, \cdots C_{mn} \end{pmatrix} = HY \tag{2.10}$$

where H is the normalized Hadamard matrix of order mn, with

$$H^T \cdot H = I.$$

Statistical analysis can be applied to the vector Z. Using subpicture size of 4 x 4, empirical analysis has revealed that the variance of C_{11} is the highest and is in excess of 10 to 1 in ratio to the next highest variance [2].

The remaining components have relatively small variances. By coarsely quantizing these C_{ij}'s close to their mean values (note: $E[C_{ij}] = 0$ for $i \neq 1$ or $j \neq 1$), due to their small variances, the resultant expected mean square error of the reconstructed video signal using these approximations should be small. This method though theoretically sound, yet, due to the fact that the visual sensation is quite different from the mean square error criterion, to improve this encoding method it requires an adequate selection of quantizations. In [1] the psychovisual property that the human eyes possess a logarithmic response was used to obtain a set of quantizations. The logarithmic quantization is chosen as follows. Let the number of representation values be $N_1 + N_2 + 1$, corresponding to N_1 positive values and N_2 negative ones. Let the resultant code corresponds to the set

$$\{-N_2, -N_2+1, \ldots, -1, 0, 1, 2, \ldots, N_1-1, N_1\}.$$

Then the inverse quantization values are chosen as follows:

$$V_k = -A_1(e^{-B_1 k} - 1) \text{ for } k = 0, -1, \ldots, -N_2$$
$$\text{and} \quad V_k = A_2(e^{B_2 k} - 1) \text{ for } k = 0, 1, 2, \ldots, N_1 \tag{2.11}$$

and the cutpoints are chosen as the $N_1 + N_2$ adjacent arithmetic means of the sequence:

$$\{V_{-N_2}, V_{-N_2+1}, \ldots V_{-1}, 0, V_1, \ldots V_{N_1}\}.$$

This quantization scheme allows the representation of zero. In addition, errors are allowed to increase exponentially.

The constants A_1, A_2, B_1, and B_2 (all positive real) determine the graininess and the maximum (or minimum) value of the quantization. Thus, quantization for a particular component is determined once the number of levels is allocated.

In addition, the d.c. component is encoded using the DPCM method, which has the following advantages:

a) Smooth transition of greylevels between adjacent subpictures, which is characterized by small video amplitude variations, can be achieved with fewer bits than that would otherwise be required. The graininess or smallest quantum jump is given by

$$A_1(e^{B_1} - 1) \quad \text{for } - \quad \text{(i.e. light to dark)}$$

and (2.12)

$$A_2(e^{B_2} - 1) \quad \text{for } + \quad \text{(i.e. dark to light)}$$

b) When the greylevel transition between adjacent subpictures is high, this is approximated logarithmically by the quantizer. Due to the logarithmic response of the human visual system, the corresponding visual sensation error is low.

c) The logarithmic DPCM coding method can correct a step function to within the graininess given by (2.12) in at most M steps, where M is a logarithmic function of the amplitude of the step function and the graininess. This is illustrated in Figure 2.1.

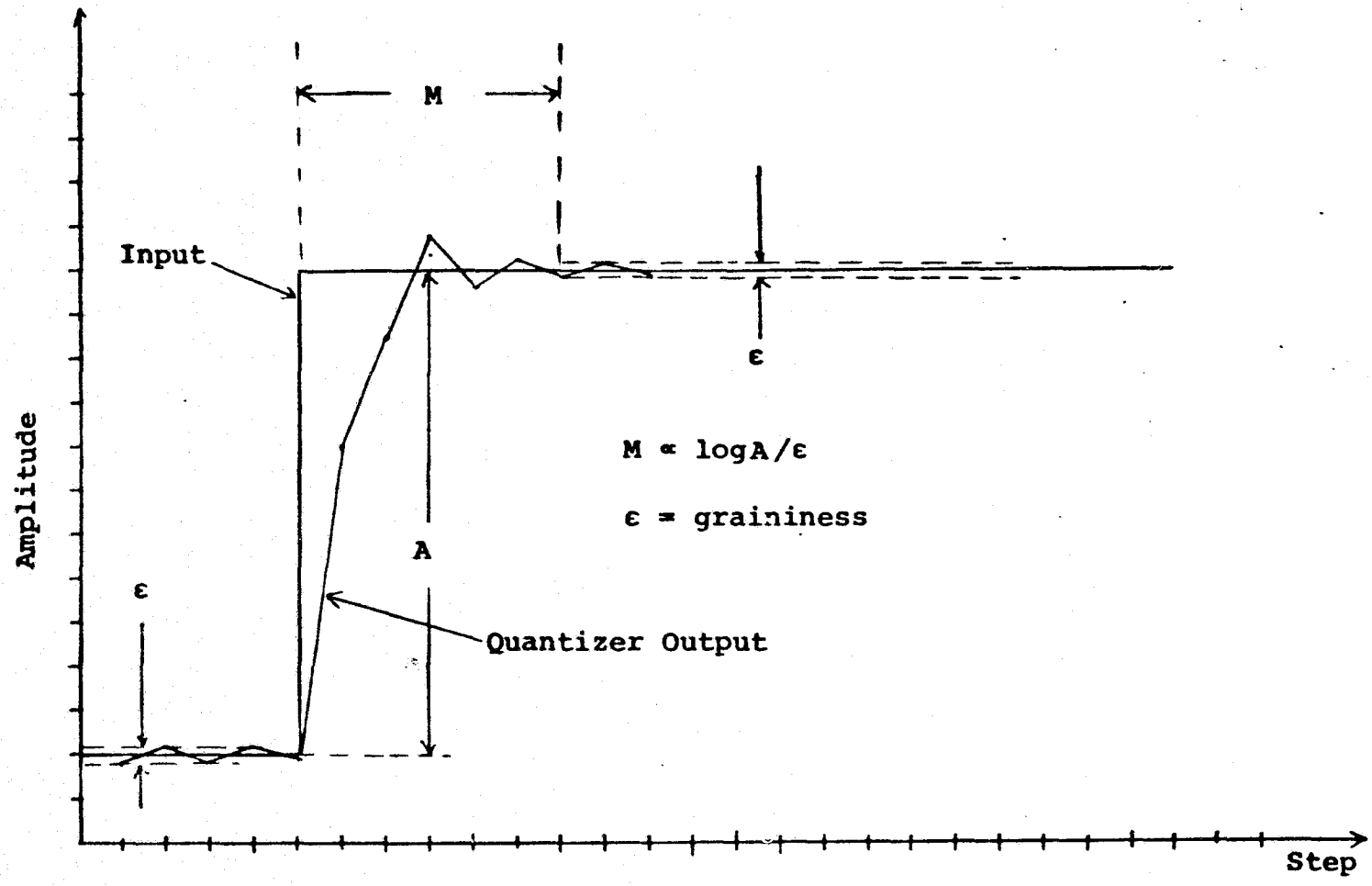


Figure 2.1. Typical Response of Logarithmic DPCM to Step Error

The non-d.c. components are not DPCM encoded for the reason that these components are generally uncorrelated between adjacent subpictures, and the fact that DPCM is inherently sensitive to channel noises.

The quantizations for various Hadamard components, (using a 4 x 4 subpicture size, see Figure 2.2), are given in Table 2.1 where

C_{11} is quantized by 32 levels (i.e. 5 bits)

C_{12} and C_{21} are quantized by 7 levels

C_{13} and C_{31} are quantized by 15 levels

C_{14} and C_{41} are quantized by 9 levels

C_{12}, C_{13} and C_{14} (likewise, C_{21}, C_{31}, and C_{41}) share 10 information bits.

C_{33}, C_{34} and C_{43} are quantized by 5 levels, and they share 7 information bits.

The overall bit requirement for the two-dimensionally compressed picture is 32 bits per subpicture, or 2 bits per picture element.

C_{11}: DPCM Logarithmically by 5 Bits (Range: $0 \leq C_{11} \leq 1024$)

Cutpoints	Representative Value
	876
761	
	646
562	
	476
414	
	350
304	
	258
223	
	188
163	
	138
118	
	100
85	
	71
61	
	50
42	
	36
29	
	23
19	
	15
11	
	8
6	
	4
2	
	0
-2	
	-4
-6	
	-8
-11	
	-14
-18	
	-22
-27	
	-32
-39	
	-47
-56	
	-65
-78	
	-91
-107	

Table 2.1. Quantization Table

C_1: DPCM Logarithmically by 5 Bits (Continued)

Cutpoints	Representative Value
	-124
-147	
	-169
-199	
	-229
-209	
	-310
-363	
	-417
-488	
	-560
-655	
	-750
-877	
	-1000

Non-d.c. components have range: -512 to +512

C_{13} and C_{31}: Quantized by 15 Levels

Cutpoints	Representative Value
	±150
±122	
	±94
±76	
	±59
±47	
	±35
±28	
	±20
±15	
	±10
±7	
	±4
±2	
	0

Table 2.1. Quantization Table (Continued)

C_{14} and C_{41}: Quantized by 9 Levels

Cutpoints	Representative Value
	±70
±53	
	±36
±26	
	±17
±11	
	±6
±3	
	0

C_{12} and C_{21}: Quantized by 7 Levels

Cutpoints	Representative Value
	±60
±43	
	±26
±17	
	±9
±4	
	0

C_{33}, C_{34} and C_{43}: Quantized by 5 Levels

Cutpoints	Representative Value
	±50
±33	
	±15
±8	
	0

Table 2.1. Quantization Table (Continued)

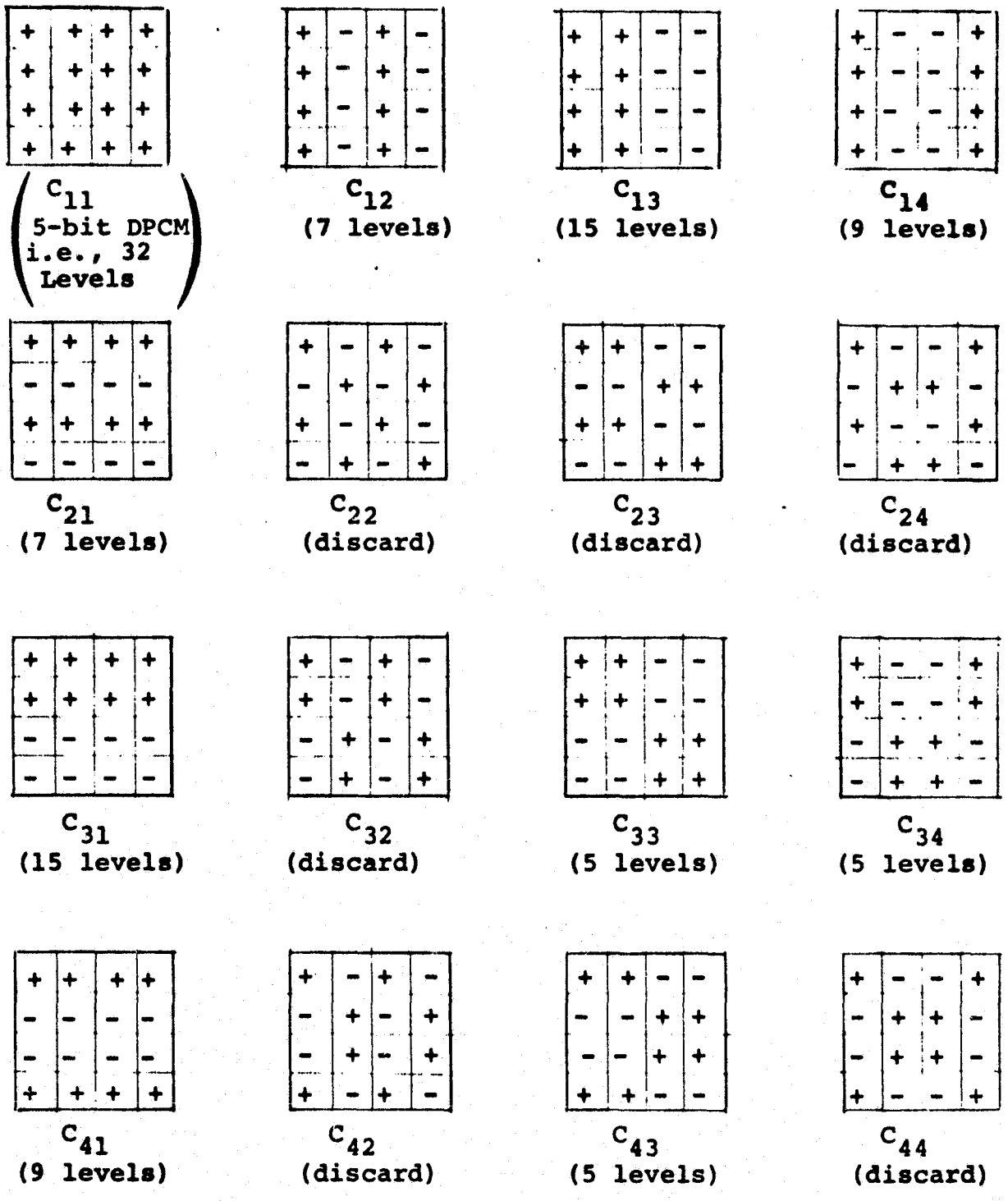


Figure 2.2. **Hadamard Component Designation**

2.2 Variable Rate, Motion Detecting Video Data Compression

Monochrome video data compression, without utilizing the time statistical redundancy over the ensemble of recognizable picture sequences, cannot operate below 2 bits per pel other than by sacrificing the picture quality substantially. Fixed rate video data compression, using frame-to-frame differencing, was studied in [1], where a compressed data rate of 1 bit per pel was achieved. To achieve a higher data compression ratio, it is necessary to use the fact that most objects in a recognizable picture sequence are stationary. Further, the human visual system tends to lose the object detail if the rate of displacement is high, and consequently its reproducing fidelity criterion can be relaxed. From the coding point of view, efficiency can be greatly increased if the number of information bits devoted to stationary objects is minimized. To achieve this, frame memories are required both at the encoder and the decoder. These are used to preserve the information of the preceding frame. At the decoder, the frame memory is used for display purposes. While at the encoder, it is used to provide decision whether to transmit updating data or not. Ideally, the decision to transmit the updating information should coincide with the object movement or mutation of the reproducing scene. Such a decision can be categorized as "motion detection."

The most primitive form of the video compression algorithm using motion detection is the "variable length, frame-to-frame difference" algorithm studied by Candy et.al [3], where the digitized video signal of each sampling point is compared with its predecessor. If the difference exceeds a certain pre-chosen threshold, then the updating information is transmitted. This scheme does not utilize the advantage of spatial statistical redundancy of the set of recognizable pictures. In addition, due to the large number of sampling points per frame, to implement this scheme, a large number of information bits must be dissipated for the "position markers." Position markers are subcodewords used to instruct the encoder and decoder where to update the information. The resultant coding efficiency is not very attractive for reasonably active picture sequences.

We shall describe a new motion detection algorithm that seems suitable for picture communication purposes, in particular, where a high data compression ratio is required for reasonably stationary picture sequences.

In the human visual system, under a well-lit situation, object movements or mutations are characterized by light intensity changes within small regions of the frame (for color picture sequences, chrominance change should also be taken into account). Hence, it is reasonable to partition the frame into disjoint regions. Again, as in the two-dimensional data compression case, the rectangular shape consti-

tutes a reasonable choice. Each region, which will be referred to as a "motion region," represents the minimum size of detectable object mutation.

Smaller motion region corresponds to finer motion resolvability, but at the expense of more information bits which must be used for addressing information. An object mutation occurring in a motion region can be defined as the deviation of light intensity from its predecessor beyond a pre-chosen threshold. (Ideally the motion detection should take into account all psychovisual phenomena, such as constant brightness, so that certain change situations, such as uniform illumination changes, will not be interpreted as motion. Again, such an "intelligent" motion detecting machine is inevitably complex).

Let the motion region be a rectangle of m x n samples:

$$Y = \begin{vmatrix} x_{11}, \cdots x_{1n} \\ \vdots \\ x_{m1}, \cdots x_{mn} \end{vmatrix} \quad m \times n \text{ vector} \qquad (2.13)$$

Then the light intensity of the region, normally defined as energy radiated per unit area, is given by

$$I(Y) = \frac{k}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^2 = \frac{k}{mn} Y^T \cdot Y, \qquad (2.14)$$

where k is a constant.

Although (2.14) is a simple arithmetic function, it is suspected that the effective intensity can be greatly simplified. This follows from the reasoning that, at normal observation distance from the displaying picture tube, the visual response to intensity and detail diminishes towards higher spatial frequencies. This leads us to suspect that the best way to estimate the intensity is to decompose (2.14) spatially (in particular, by Hadamard transformation) and note that if Y is transformed into Hadamard components:

$$Z = \begin{pmatrix} C_{11}, \cdots C_{1n} \\ \vdots \\ C_{m1}, \cdots C_{mn} \end{pmatrix} = HY$$

m x n vector

then

$$Y^T \cdot Y = Z^T \cdot Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C^2_{ij} \qquad (2.15)$$

Consequently, it is natural to decompose I(Y) into

$$I(Y) = \sum_{i=1}^{m} \sum_{j-1}^{n} I_{ij}(Y) \qquad (2.16)$$

$I_{ij}(Y)$ is the intensity contributed by the spatial frequency corresponding to the pattern given by C_{ij}. It is suspected that the intensity is mainly concentrated on the d.c. component. This can be shown as follows:

The expected intensity for the motion region is

$$E[I(Y)] = \int_{\Omega} I(Y)\ p_Y(\omega)\, d\omega$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} E[I_{ij}(Y)]$$

where

$$E[I_{ij}(Y)] = \frac{k}{mn} \int_{\Omega} C_{ij}^2(\omega)\ p_Y(\omega)\, d\omega$$

$$= \frac{k}{mn} \left[\mathrm{var}(C_{ij}) + \overline{C_{ij}}^{\,2} \right]$$

where $\overline{C_{ij}}$ = mean value of C_{ij}

= 0 for $i \neq 1$ or $j \neq 1$

$\overline{C_{11}}$ can be estimated by assuming that C_{11} uniformly distributes over the range of C_{11}, i.e.

$$0 \leq C_{11} \leq E_m = 255\sqrt{mn}\ .$$

From the assumption of uniform distribution, we have

$$\overline{C_{11}} = E_m/2$$

$$\mathrm{var}\ (C_{11}) = E_m^2/3 - E_m^2/4 = E_m^2/12$$

Thus, we have

$$E[I_{11}(Y)] = \frac{k}{3mn} E_m^2$$

$$E[I_{ij}(Y)] = \frac{k}{12mn} \xi_{ij}\ E_m^2 \qquad i \neq 1 \text{ or } j \neq 1,$$

where $\xi_{ij} = \mathrm{var}\ (C_{ij})/\mathrm{var}\ (C_{11})$.

For the 4 x 4 Hadamard square, when the statistics are taken over the ensemble of recognizable subpictures, using the

typical statistics obtained by Landau and Slepian, i.e., Table 2.2,

(i,j)	ξ_{ij}	(i,j)	ξ_{ij}
(1,2)	.051	(1,4)	.035
(2,1)	.048	(4,1)	.038
(2,2)	.014	(2,4)	.016
(1,3)	.098	(4,2)	.015
(3,1)	.087	(3,4)	.024
(2,3)	.022	(4,3)	.024
(3,2)	.02	(4,4)	.019
(3,3)	.034		

Table 2.2

we have $E[I(Y)] = E[I_{11}(Y)] + \frac{.545}{4} E[I_{11}(Y)]$, where the second term is the total expected intensity contributed by the non-d.c. components, which is about 12% of the total expected intensity. Note, however, that if we assume the distribution of C_{11} concentrates about its mean, as in the case of a normal video reproduction, then the expected intensity will further concentrate on the d.c. component. It is also interesting to note that intensity estimation using C_{11}, C_{13} and C_{31} contributes 93% of the total expected intensity.

The above leads us to believe that the intensity approximation using the d.c. component (or few additional components) is sufficient for motion detection purposes. Motion detection in the Hadamard coordinates has the additional advantage that the two-dimensional data compression obtained previously can be applied directly.

Thus, our motion detection algorithm consists of:

1) The size of motion region is selected to be one or an integral multiple of Hadamard subpictures. If the motion region contains L subpictures, then the effective intensity of the motion region is defined as

$$I = \frac{1}{L}\sum_{j=1}^{L} C_{11}^2 \ (j)$$

where $C_{11}(j)$ is the d.c. component of the j-th subpicture.

2) A reference frame is transmitted using the two-dimensional compression technique given in 2.1.

3) At the encoder, a frame of memory is used to store the information of the d.c. components. The decoder frame memory has sufficient storage for the d.c. components and the encoded non-d.c. components.

4) For the subsequent frames, the intensity of each motion region is compared with its predecessor. When the difference exceeds a pre-chosen threshold, the difference of the d.c. component is quantized by 5

information bits (using the same quantization table as in (1)). This along with the new compressed non-d.c. components and an address code are transmitted. Simultaneously, the frame memory of the encoder is updated.

5) At the receiving end, the address code received directs the decoder to the portion of the frame memory where updating is to be performed.

6) Visual display is accomplished by inverse quantization lookup, inverse Hadamard transformation, and D-to-A conversion of the data stored in the decoder frame memory.

Experiments using this algorithm have been performed on the LIM Video Controller at LINKABIT. In particular, the motion region was chosen to be one subpicture; the resultant motion detection simply reduces to a comparison of the d.c. components. The overall subjective picture quality was judged to be good. In particular, stationary objects were faithfully reproduced. This feature makes it very attractive for both normal and inactive scenes.

Coding using the motion detection method requires position markers to instruct both the encoder and decoder where about to update the data. The position markers should be capable of addressing every motion region. (Note: due to the fact that the number of motion regions is substantially less than that of sampling points, the number of information

bits dissipated in the position marker is greatly reduced). There are many different ways to implement the position markers. The following is a list of possible encoding schemes:

1) The most primitive scheme is to each updating information a code of constant length is assigned, which consists of two portions:

← 14 bits address →	← 32 bits updating content →

There are (512/4) x (480/4) = 128 x 120 motion regions (in the sequel, we have assumed one motion region = 1 Hadamard square, i.e., 4 x 4 samples), which requires 14 information bits. The overall compressed rate operates between the following limits:

$0 <$ compressed rate $\leq 2 + \frac{14}{16}$ bits per pel.

When properly synchronized by a frame synchronizing code this method has the effect of being less sensitive to channel noises.

If, in addition, a horizontal synchronizing code is provided for each horizontal line group, then the position marker requires only 7 information bits. The corresponding compressed rate operates between the following limits:

A bit/pel $\leq$ compressed rate $\leq 2 + \frac{7}{16}$ + A bits/pel

where A is the efficiency loss due to the synchronizing codes.

2) Updating information can be coded using a variable length subcodeword, which contains the address information, a set of updating data, representing a number of consecutive motion regions that need updating. The length of this subcodeword can be determined either by an end marker, a continuation marker, or a number represented by a fixed number of information bits:

a)

|← address →|← 32 bit data →|← 32 bit data →|...|← 32 bit data →|end marker|

b)

|← address →|← 32 bit data →|c|← 32 bit data →|...| 32 bit data |c|

where c is the continuation marker represented by a fixed number of information bits.

c)

|← address→|← N →|← data | ... | data |

N data

This method is particularly efficient for picture sequences with fairly active scenes. However, it is quite sensitive to channel noises. For example, erroneous diciphering of the endmarker, or continuation marker, or the number of consecutive data, N, will result in errors for the subsequent decoding. In this case, a synchronizing code, preferably for each horizontal group, must be used to minimize the accumulative effect of channel error to limited areas.

3) Each motion region is coded by a motion indicator using one information bit. For example, 1 represents motion, and 0, no motion. If 1 occurs, then it is followed by the 32 bits of updating information. The corresponding compressed bit error rate versus the amount of motion detected is illustrated as curve A of Figure 2.3. This method is also sensitive to channel noises, and a provision of line group synchronizing code is desirable.

4) The motion indicator can be implanted in the subcodeword of the content information allocated for the d.c. component. The d.c. component is coded by 5 information bits. During motion detection, the difference of the d.c. components are quantized and transmitted. If the difference, Δ, has absolute value greater than the prechosen threshold, then a subcodeword of length 27 bits, which contains the updating information for non-d.c. components, is also transmitted. Otherwise, only the quantized difference is transmitted.

5 bits	27 bits
value Δ d.c. component	subcodeword for non-d.c. components send only if $\lvert\Delta\rvert \geq$ threshold

In this case, the compressed bit rate versus the amount of motion detected is illustrated as curve B of Figure 2.3.

Again, this method is sensitive to channel noise and a provision of line group synchronizing code is desirable.

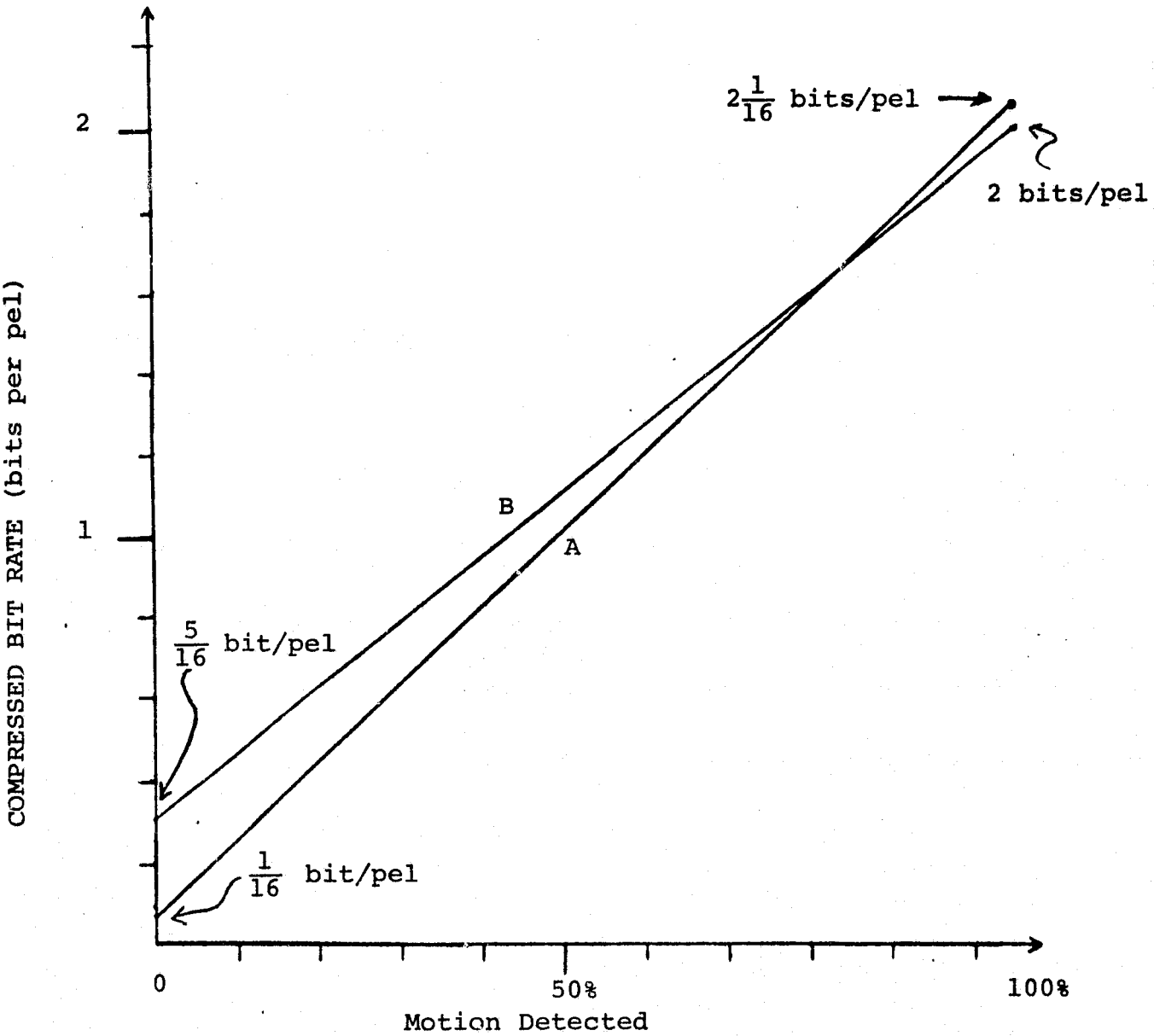


Figure 2.3. Compressed Bit Rate vs. Motion Detected

In addition, due to the fact that the d.c. components are transmitted in differential mode, reference frames should be sent every so often to reduce the accumulative effect of channel errors. (This cumulative error effect only affects each motion region individually.)

2.3 Transforming the Variable Rate Algorithm into a Fixed Rate Algorithm

The motion detection algorithm described in the above section has the distinct disadvantage of being variable in coding length, and thus dependent entirely on the motion activity. To transmit the data, a rate buffer must be used. Since, under arbitrary situations, the duration of active scenes may vary widely, a very large buffer may be needed to reproduce every frame of an arbitrary picture sequence. This report proposes an adaptive rate buffer scheme that seems most suitable for the up-link Space Shuttle TV comunication system. This scheme is now described:

At the encoder, there are two identical rate buffers. Each buffer has sufficient storage to store one field of encoded data (i.e., both updating data and position markers). These buffers are used to store field 1 and field 2 of the encoded data. Data are transmitted in blocks of one field. The buffers can be in one of the following modes:

1) Active Mode: The buffer is being used to transmit the data,

2) Dummy Mode: The buffer is being filled by the source data,

3) Inactive Mode: No data are being taken or filled.

The field 1 and field 2 buffers alternate between active and dummy modes with a possible inactive mode in between. Thus, under normal conditions, one of the buffers whose contents are being transmitted while the other is being filled. Due to the variable length of the data queues, the following events may occur:

1) The dummy is filled but the active buffer is not emptied. In this case, no source data is accepted. Furthermore, the video data compression process is stopped. This is done by halting the A-to-D converter, so that the frame memory of the source encoder (which is used to store the d.c. components) contains only information of the last processed fields. This halting condition prevails until the active memory is emptied; then the dummy buffer is switched to active mode, and the emptied buffer becomes the dummy. The dummy accepts only the data of the next field in sequence (i.e., if the last active buffer is field 1, then the dummy accepts the data of the following immediately available field 2). Thus, at most one frame time may elapse between the last data taken from the active buffer and the first data to be entered into the dummy buffer. This can be done by reactivating the A-to-D converter and the data compression process at the beginning of the next field in sequence.

2) The active buffer is emptied, and the dummy is not filled (Note: it takes exactly one field time to complete the filling, i.e., the input of the rate buffer accepts, when activated, source data at a real time rate), then the active buffer is made inactive. Similarly, if the active buffer is emptied and the dummy is waiting to be filled (this circumstance occurs when the next field in sequence, described in (1), has not yet arrived), then no data are being taken or filled. During these periods, special codewords are transmitted representing "blank" or "waiting situation," (or the time can be used to transmit field synchronizing codewords.)

In doing so, data blocks of alternating fields are transmitted such that the beginning of each data block always occurs at about the beginning of the realtime field timing. The periods in between are filled with blanks. Many fields may not be processed if the picture contents are very active.

In addition, special codewords are transmitted at the beginning of each field to identify whether the data following belong to field 1, field 2, reference data or updating data.

At the receiver, no additional buffer memories are needed. Again, the frame memory organization is divided into field 1 and field 2. The memories store the decoded d.c. components and the encoded

non-d.c. components. One of the field memories is always in "display" mode, i.e., the data are taken, inverse quantized, inverse Hadamard transformed, D-to-A converted and displayed on the TV monitor, while the other field memory is either being updated or filled with reference data. If the updating, or filling with reference data, is not completed at the end of the normal TV field scan, then the contents of the current displaying field are repeated for the following two fields. This condition prevails until the entrance of data is completed at the end of the TV field scan corresponding to the designated field of the displaying memory. Then the roles of the field memories are exchanged. In doing so, the TV monitor displays a sequence of pictures that consists of alternate bursts of the two fields. Each burst has odd number length:

Field 1 ... Field 1, Field 2 ... Field 2 ....

odd number odd number

In particular, if the source data rate is lower than the data transmission rate, then the burst lengths become 1, i.e., the displayed picture sequence coincides with the input picture sequence.

When the source data rate exceeds the transmission rate, the effect is that the input picture sequence is abridged, by discarding information in groups of consecutive frames, where the contents become active. This allows the transmission data rate to catch up with the source data rate. Likewise, if the source data rate is less than the transmission rate, then blanks are transmitted to fill up the gaps between the source and transmission. The overall visual effect is that the transient details of object movements or mutations are abridged, which is not very objectionable from the picture comprehensibility point of view.

This method is well suited for the space shuttle uplink TV communications system for the following reasons:

a) No additional rate buffer memories are required at the decoder, which is located at the space vehicle where the envionment is limited. Consequently, the augmentation of hardware complexity is minimal.

b) The rate buffer is only needed at the encoder, which is located at the ground station, where the increase in hardware complexity does not affect the feasibility of the system.

c) Due to the nature of the very low information rate for the uplink TV communication, the advantage of

time statistical redundancy (in the form of stationary objects) is utilized maximally. In particular, when the source rate is less than the transmission rate (for 1 Mbits per second uplink rate, this corresponds roughly to motion detected over 6% of the frame), the picture sequence is reproduced without abridging. When an inevitably large amount of object motion or mutation occurs this is remedied by abridging the input sequence. Note the maximal abridging effect for 1 Mbits per second corresponds to displaying the alternate fields at approximately 1/4 second intervals. This corresponds to 100% motion detected.

The functional block diagrams in the uplink video source encoder and decoder are illustrated in Figures 2.4 and 2.5. The field rate buffers mentioned above, due to the random length nature of motion detected, are implemented by random access memories. As mentioned previously, these accept data (i.e., both encoded data and/or position markers) at a real time rate.

The rate buffer described above is adaptive in the sense that its input is controlled by its output. This is shown in Figure 2.5, where an "end-of-field" decoder is used to sense the end of information taken from the buffer and to signal the controller to swap the roles of the rate buffers. This decoding function does not necessarily provided codeword

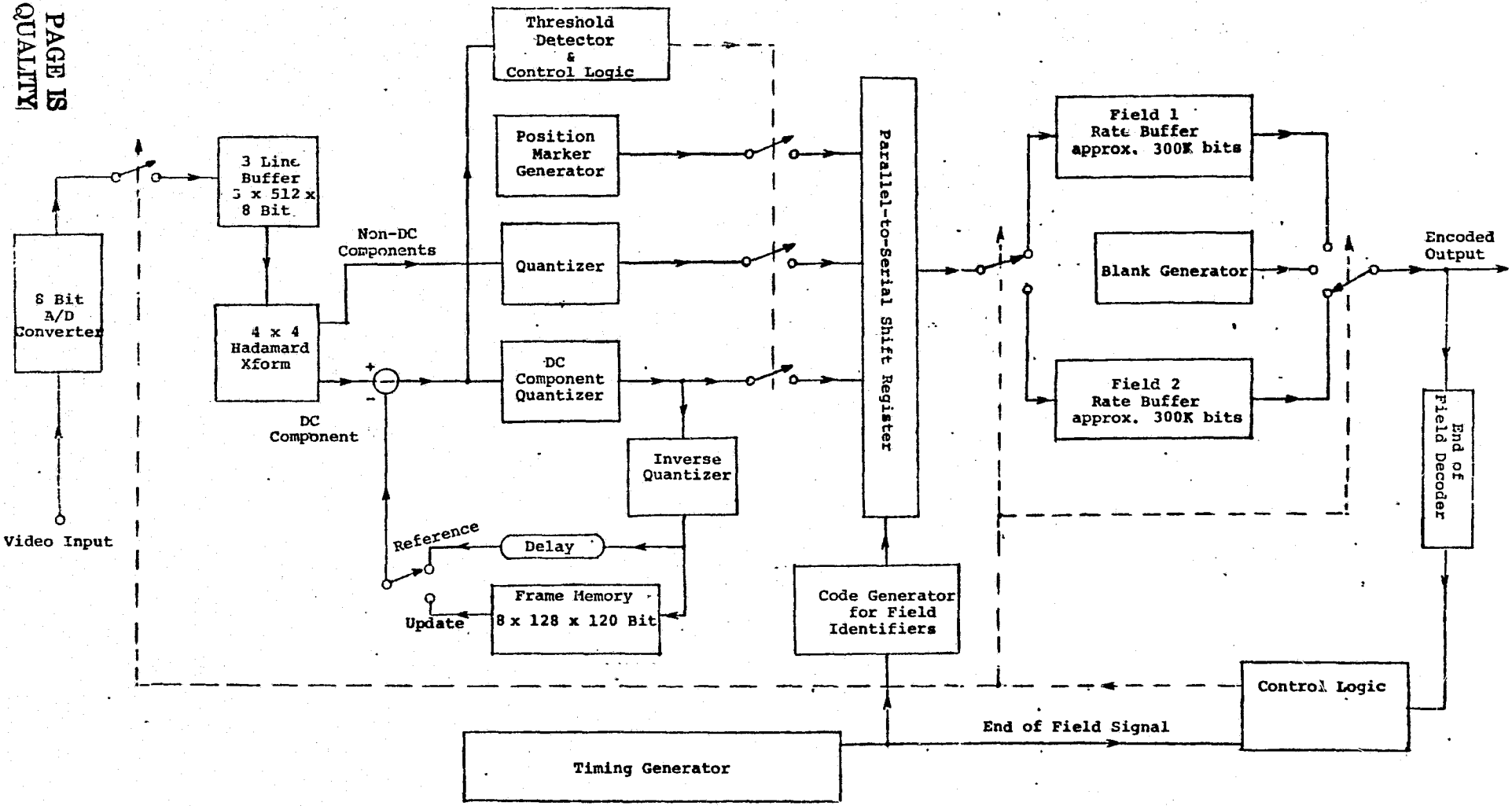


Figure 2.4. UPLINK VIDEO SOURCE ENCODER

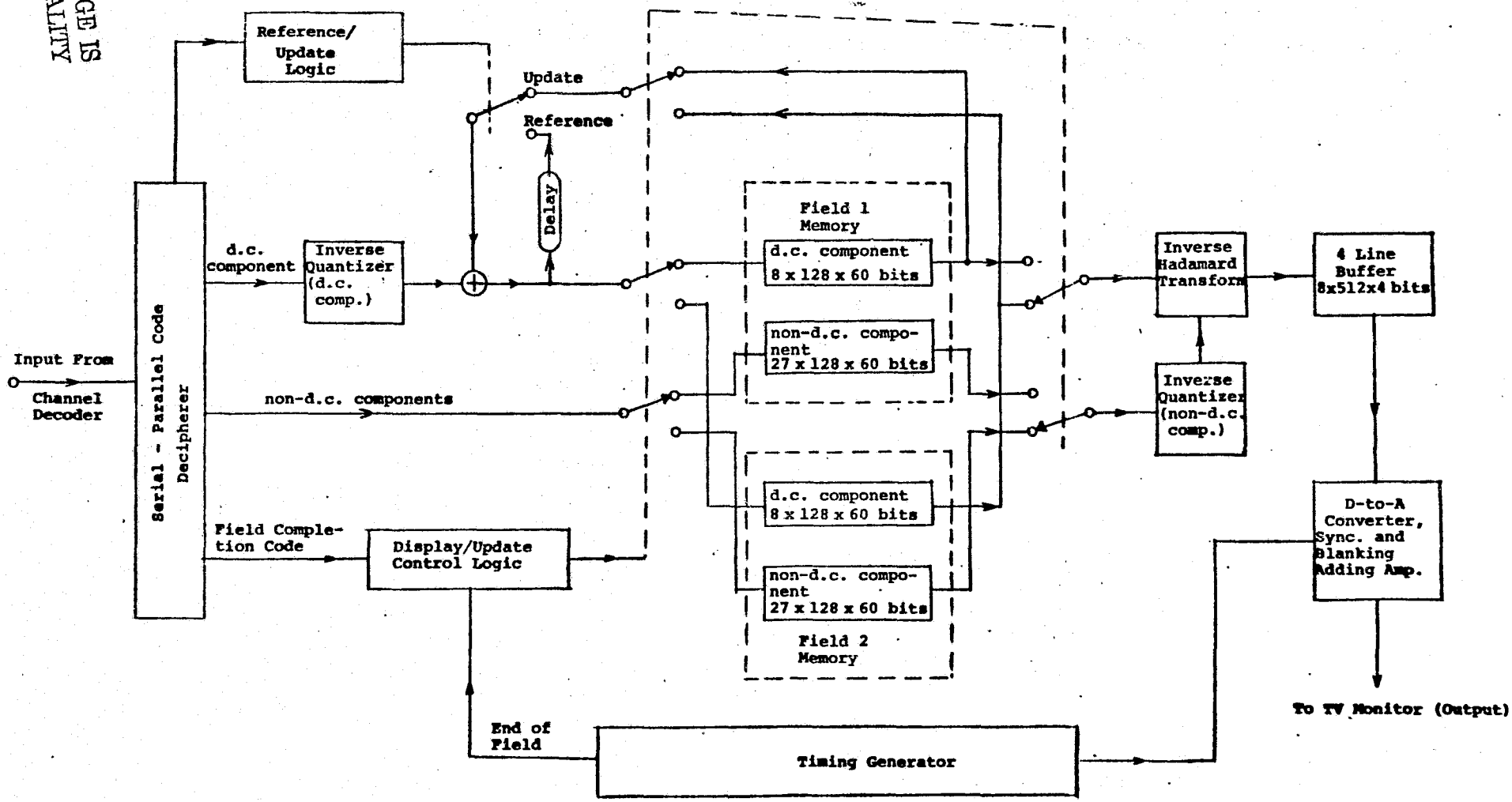


Figure 2.5. UPLINK TV SOURCE DECODER

deciphering, in the usual sense, where complicated hardware may be required. For example, this can be accomplished by using a content counter for each field rate buffer. The counters keep track of the amount of data entered during the updating mode. In the active mode, the counters keep track of the amount of data transmitted. A typical hardware implementation for the rate buffer is shown in Figure 2.6, where a random access memory of size 299.52 Kbits is used. The memory is divided into two parts: data memory and motion address memory (special synchoronizing codewords are used for each horizontal line group). During updating (or entering reference data), data are entered in blocks of 32 bits for the compressed data and 7 bits for the motion address memory (used only in the updating mode). When the data compressor detects the first motion, the motion region address is entered followed by the updating data. Simultaneously, the buffer counter is incremented, (at the beginning of the updating mode, it is reset to 0). If motion is detected on the motion region of the next successive address, then a continuation marker is entered to the motion address memory. In other words, motion address data is entered only if the motion region is the leader of a burst of consecutive updating motion regions. Data enters the rate buffer sequentially. During the active mode, data is taken out bit by bit at the transmission rate. If a continuation marker is encountered, then the

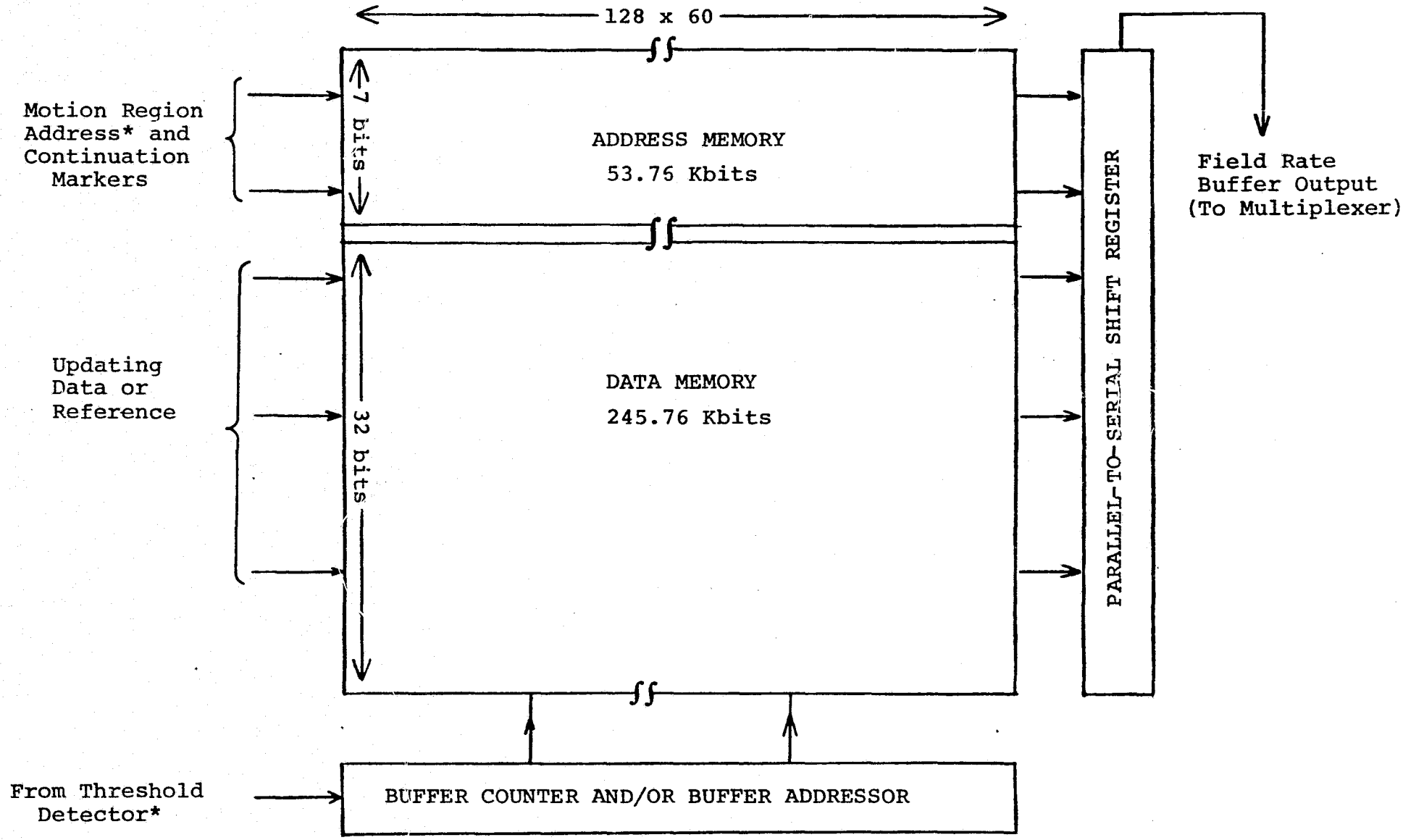


Figure 2.6. Typical Hardware Organization of a Field Rate Buffer

address code is ignored. If not, then an end-marker is transmitted and this is followed either by the location address of updating motion regions, or by the special end-of-line synchronizing codeword. The buffer counter is decremented each time a block of data is transmitted. End of active mode is signalled by the 0 content of the buffer counter.

3.0 Color Video Source Coding

We shall begin this section with the fundamental principles of color TV communication and then proceed to devise a feasible digital color video data compressor suitable for the down-link space shuttle TV communication system. Color picture reproduction relies mainly on the principle of three primary color decompositions, where the picture is sampled through a red, green, and blue filter. Hence, one may treat a color signal as a vector of three separate monochrome signals corresponding to the red, green, and blue contents of the object picture. If the color signal is transmitted in this manner, it would require three times the bandwidth needed for the monochrome transmission. As in the monochrome case, statistical redundancies exist not only as the spatial or time statistical correlations for each color, but also in the form of intra-color redundancy. Data compression can be accomplished in three separate steps. First, the bandwidth can be reduced due to the intra-color redundancy, then the spatial and time statistical data reduction described in Section 2.0 can be applied directly. The intra-color statistical redundancy can be described by considering the random vector,

$$S(t) = \Big(R(t),\ G(t),\ B(t)\Big)^T,$$

where $R(t)$ = red video signal

$G(t)$ = green video signal

$B(t)$ = blue video signal,

as a continuous random process. Bandwidth reduction, in the sense of the least expected mean square error criterion, can be obtained by applying the Karhunen-Loeve procedure to the set of recognizable color pictures. This normally results in a linear transformation of the original color signal vector,

$$\begin{pmatrix} K_1(t) \\ K_2(t) \\ K_3(t) \end{pmatrix} = M \begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix}$$

where M is a 3 x 3 matrix that diagonalizes the covariance matrix

$$E[S(t)^T \cdot S(t)].$$

This Karhunen-Loeve procedure depends mainly on empirical analysis. Psychovisual phenomena are not invoked. Other transformations were sought. One of the methods used in the standard NTSC color TV transmission is to transform the signal vector S(t) into the coordinates that consist of the monochrome brightness and two vectors that lie on the chrominance plane, see Figure 3.1.

$$\begin{pmatrix} Y(t) \\ I(t) \\ Q(t) \end{pmatrix} = M \begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix}, \tag{3.1}$$

where

$$M = \begin{pmatrix} .3 & .59 & .11 \\ .6 & -.28 & -.32 \\ .21 & -.52 & .31 \end{pmatrix} \tag{3.2}$$

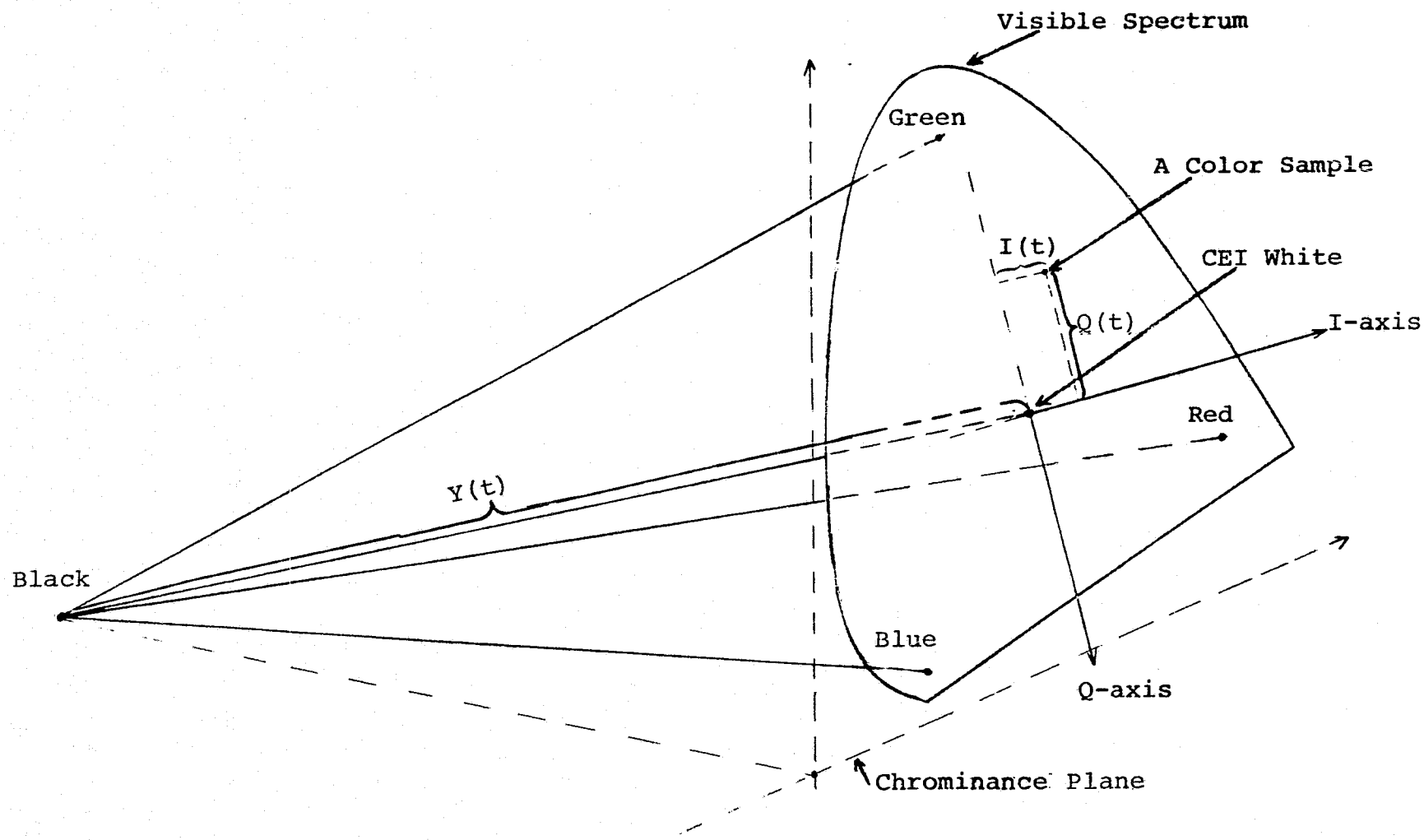


Figure 3.1 Transformation of Color Sample into Y, I and Q Coordinates

Y(t) corresponds to the monochrome brightness, I(t) corresponds to the vector in the "orange red-cyan" direction in the chrominance plane, and Q(t) corresponds to the vector in the "magenta-green" direction. This choice of coordinates has the following advantages:

1) Black and white monochrome video signal can be readily formed by simply dropping the Q and I components,

2) Psychovisual phenomenon that the human eyes sense only black and white at very low luminosity,

3) For small color areas, the human eyes exhibit "tritanopia" (or two color vision). This corresponds to the decrease of spatial visual sensitivity in the Q component.

Due to these facts, color video signals can be transmitted, with reasonable picture quality, when Y(t) has a bandwidth of 4 MHz, I(t) has a bandwidth of 1.5 MHz, and Q(t) has a bandwidth of only .5 MHz. Standard commercial color TV signals are transmitted by modulating the chrominance signals, I(t) and Q(t), by a subcarrier ω_c:

$$\begin{aligned} M(t) &= I(t)\ \mathrm{Sin}\ \omega_c t + Q(t)\ \mathrm{Cos}\ \omega_c t \\ &= C_s(t)\ \mathrm{Sin}\ (\omega_c t + C_h(t)), \end{aligned} \tag{3.3}$$

where

$$C_s(t) = \sqrt{I^2(t) + Q^2(t)} \tag{3.4}$$

and

$$C_h(t) = \mathrm{Tan}^{-1}\ Q(t)/I(t) \tag{3.5}$$

$C_s(t)$ and $C_h(t)$ correspond, very roughly,to the constant hue and constant color saturation stream lines in the plane of chrominance. Thus, a typical commercial color TV signal can be expressed as

$$Y(t) + M(t) = Y(t) + I(t) \text{ Sin } \omega_c t + Q(t) \text{ Cos } \omega_c t \quad (3.6)$$

This composite modulated signal, together with the reference phase of the subcarrier frequency, enables the receiver to demodulate Y(t), I(t) and Q(t). The corresponding primary signals, R(t), G(t) and B(t) are obtained by the inverse linear transformation:

$$\begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix} = M^{-1} \begin{pmatrix} Y(t) \\ I(t) \\ Q(t) \end{pmatrix}$$

The general color TV communication scheme is illustrated in Figure 3.2.

3.1 Digital Color Video Data Compression

Equation (3.6) enables color video information to be packed in a continuous analog signal. Therefore, the monochrome digital video data compression technique can be applied if the mean square error of the decoded signal is sufficiently small so that the phase and amplitude errors of the reconstructed signal produce tolerable visual sensational errors. This can be done by increasing the sampling rate of the A-to-D converter and assigning more information bits to the quantization of the Hadamard components. This method has been experimented by Enomoto and Shibata [4] using 3.75 bits per pel.

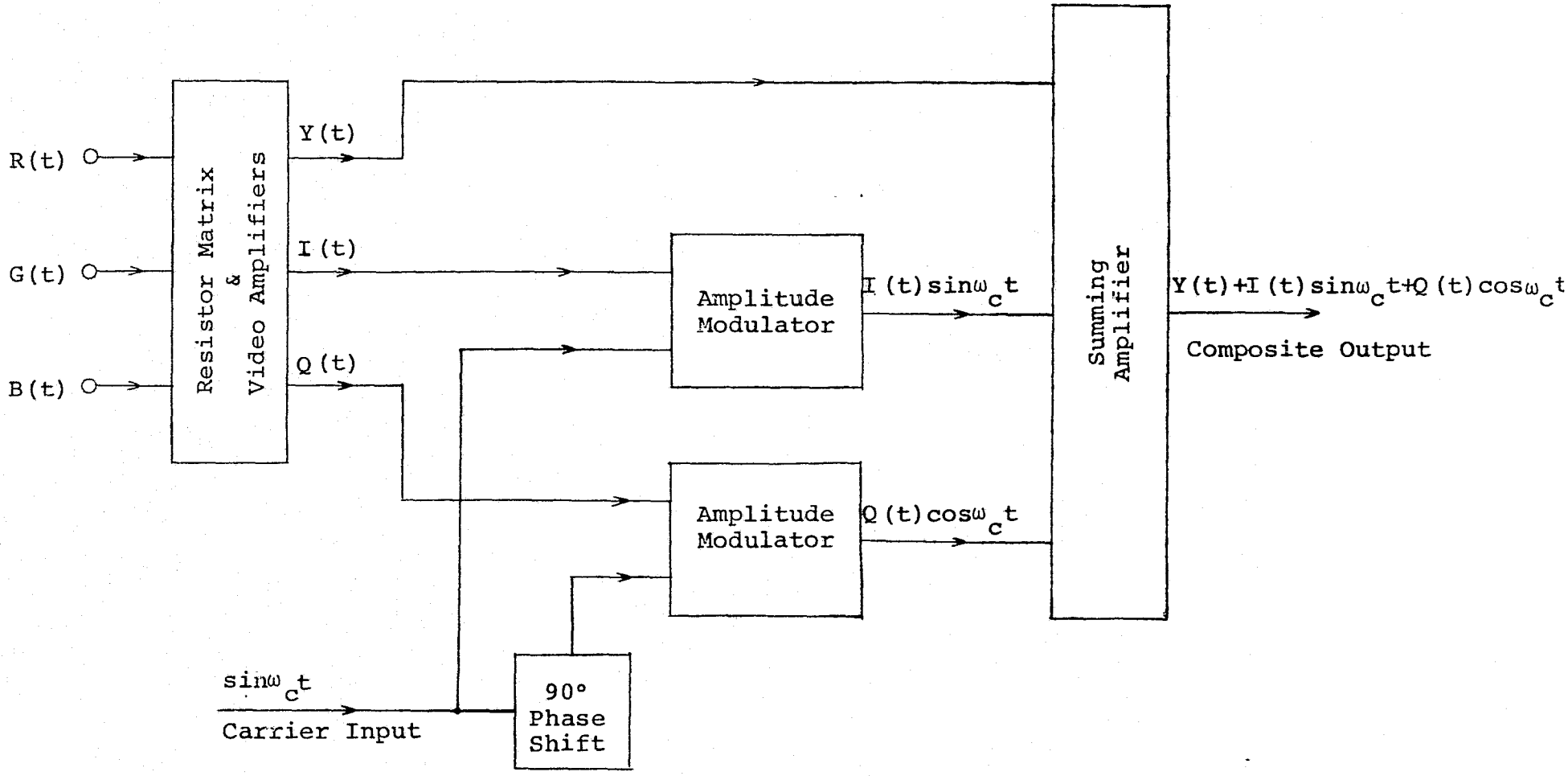


Figure 3.2. Modulated Composite Color Signal

However, this method is not recommended by this report for the following reasons:

1) To resolve the phase and amplitude information of the modulated signal to within tolerable visual sensations, a substantially higher sampling frequency must be assigned to the A-to-D converter. The typical requirement in this case is in the range of 12 MHz to 15 MHz. A-to-D converters of this kind operate in the range of the present state of the art. The corresponding data conversion accuracy, power consumption, weight and cost make them undesirable for limited environment, such as a space vehicle used in the Space Shuttle program.

2) The corresponding Hadamard transformer and quantizer must also operate at higher rates. Consequently, power consumption, weight, etc., will increase.

3) The additional analog modulation-demodulation of the chrominance signals will contribute additional errors to the overall system.

A digitized color video data compression algorithm suggested by this report is given as follows:

The three primary color video signals, R(t), G(t) and B(t) are first analog transformed into Y(t), I(t), and Q(t), using simple resistor network and video inverting amplifiers. Among these only the monochrome brightness

component, Y(t), conveys most of the picture information and requires highest transmission bandwidth. As in Section 2.0, 512 samples per horizontal line (approximately 8 MHz sampling frequency), and 8 information bits per sample are sufficient for video reproduction purposes. I(t) and Q(t) have substantially lower bandwidth requirements (this is resulted mainly from the poor spatial response of the human visual system to the chrominance components). Comparatively, either I(t) or Q(t) has bandwidth requirements of less than half of that required by Y(t). This enables us to sample I(t) and Q(t) at half the sampling frequency used for Y(t) while maintaining sufficient chrominance resolution. The reduction in horizontal sampling rate is mainly due to the loss of chrominance response of human eyes to higher spatial frequencies. This reason can be applied equally well vertically. Consequently, I(t) and Q(t) may share the same A-to-D converter by sampling I(t) and Q(t) at alternate horizontal lines. (Note: this vertical bandwidth reduction cannot be applied directly in the case of analog modulation-demodulation described in (3.1) due to the horizontal scanning nature of the TV communication system). For ripple-free operation, the sampling frequency for the A-to-D converter, used for Q(t) and I(t) is synchronized to that used for Y(t) by a simple frequency divider. In addition, since the chrominance resolution of human eyes is much less than that of monochrome

brightness, a 6-bit resolution seems to be sufficient for the I(t) and Q(t) A-to-D converter. This corresponds to 31 or more levels of color purity in the directions: white to cyan, white to orange red, white to green, and white to magenta.

Using the above, the additional encoder front-end analog circuit requirements are: a 6-bit A-to-D converter, a resistor network, and an analog multiplexer for the I(t) and Q(t) signals. For limited environments, as in a space vehicle, this seems to be more preferrable than to compress the components modulated signal, (3.6).

In doing so, the digitized color signal can be represented by a lattice of 512 x 480 sample points for the monochrome brightness component, Y(t), and a lattice of 256 x 240 sample points for the chrominance components, I(t) and Q(t). The monochrome brightness subpictures are chosen with size of 4 x 4 samples, and the chrominance subpictures with 2 x 2 samples. Larger chrominance subpicture sizes are not used because the spatial statistical correlation between sampling points within each subpicture is only a function of its physical dimensions. The bandwidth reduction obtained by lowering the sampling rate is resulted merely from the psychovisual phenomena. Further, the chrominance subpictures are made to coincide with the monochrome brightness subpictures.

This is illustrated in Figure 3.3. The corresponding digitized color subpictures are represented by the following vectors:

Brightness Vector

$$Y = \begin{pmatrix} x_{11} & \cdots & x_{14} \\ \vdots & & \\ x_{41} & \cdots & x_{44} \end{pmatrix} \tag{3.7}$$

Cyan-Orange Red Vector

$$I = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u22 \end{pmatrix} \tag{3.8}$$

and Green-Magenta Vector

$$Q = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}, \tag{3.9}$$

where x_{ij} is the sampled brightness value at the (i,j) coordinate of the subpicture, and the u_{ij}'s and v_{ij}'s are the sampled chrominance values. This is illustrated in Figure 3.3. They have integer representations in the following ranges:

$$0 \leq x_{ij} \leq 255$$

$$0 \leq u_{ij}, v_{ij} \leq 63$$

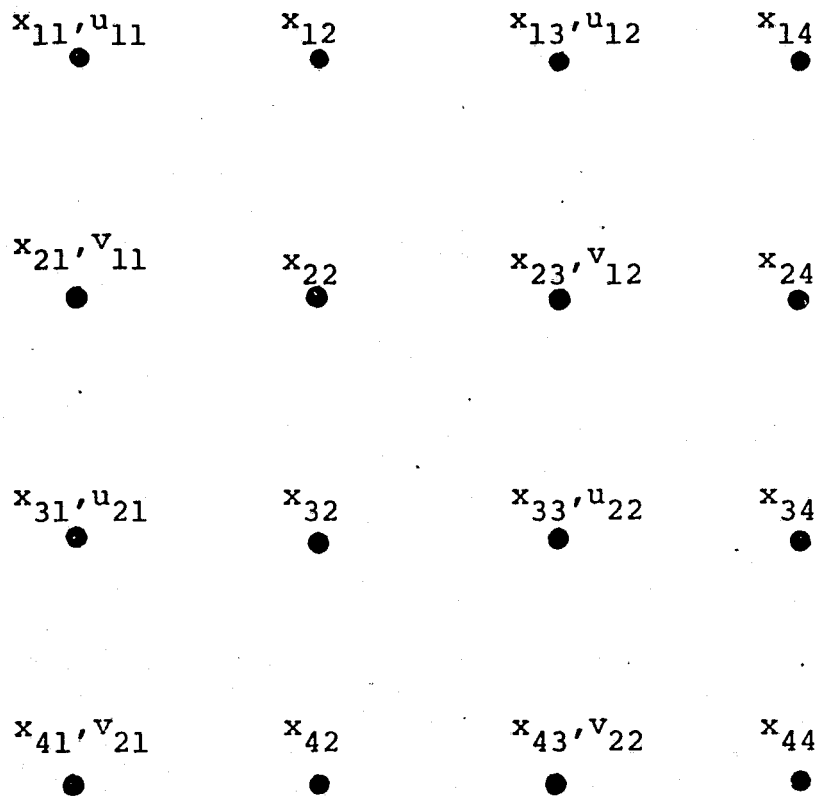


Figure 3.3. Orientations of Subpicture Sample Points

Hadamard transformations can be applied to the above vectors:

$$HY = \begin{pmatrix} C_{11}^{Y} & \cdots & C_{14}^{Y} \\ \vdots & & \\ C_{41}^{Y} & \cdots & C_{44}^{Y} \end{pmatrix} \tag{3.10}$$

$$H'I = \begin{pmatrix} C_{11}^{I} & C_{12}^{I} \\ C_{21}^{I} & C_{22}^{I} \end{pmatrix} \tag{3.11}$$

and

$$H'Q = \begin{pmatrix} C_{11}^{Q} & C_{12}^{Q} \\ C_{21}^{Q} & C_{22}^{Q} \end{pmatrix} .$$

Where H is the Hadamard transform for 4 x 4 subpictures, and H′ is that for 2 x 2 subpictures. The transformed range is given as follows:

$$0 \leq C_{11}^{Y} \leq 4 \times 255$$

$$-2 \times 255 \leq C_{ij}^{Y} \leq 2 \times 255 \text{ for } i \neq 1 \text{ or } j \neq 1$$

$$0 \leq C_{11}^{I},\ C_{11}^{Q} \leq 2 \times 63$$

$$-63 \leq C_{ij}^{I},\ C_{ij}^{Q} \leq 63 \text{ for } i \neq 1 \text{ or } j \neq 1$$

The monochrome video data compression procedures described in Section 2.0 can be applied to HY. When two

dimensionally compressed HY requires 2 bits per pel, or equivalently, 491.52 Kbits per frame. Likewise, components for H'I and H'Q can be compressed using similar logarithmic quantization procedures. In particular, due to the low spatial frequency response for the magenta-green chrominance component, C^Q_{12}, C^Q_{21} and C^Q_{22} can be discarded. This follows from the reasoning that using C^Q_{11} alone, to approximate the Q(t) component, corresponds to sampling Q(t) at 1/4 the sample frequency for Y(t), or, equivalently, sampling at 2 MHz. This is substantially higher than the Nyquist rate required for the 500 KHz bandwidth of the Q(t) signal used in standard commercial color TV signal. Logarithmic DPCM technique should be applied to the C^I_{11} and C^Q_{11} components in order to extract maximal advantage due to the residual statistical correlation existing between adjacent subpictures.

The quantization table for chrominance components is given in Table 3.1. The designation for the Hadamard components is shown in Figure 3.4.

C^I_{11} and C^Q_{21} have 23 quantization levels using logarithmic DPCM methods.

C^I_{12} and C^I_{21} have 15 quantization levels.

C^I_{22} has 9 quantization levels.

C^I_{11} and C^Q_{11} together have 23 x 23 = 529 possible pairs of representative values. To encode them, it would requires

C_{11}^{I} and C_{11}^{Q}: Quantized by 23 levels.

Cutpoint	Representative Value
	± 108
+ 96	
	± 84
± 74	
	± 64
± 57	
	± 50
± 44	
	± 38
± 33	
	± 28
± 24	
	± 20
± 17	
	± 14
± 11	
	± 9
± 7	
	± 5
± 3	
	± 2
± 1	
	0

In addition, the following cutpoint pairs of (C_{11}^{I}, C_{11}^{Q}) are deleted, these occur outside the reproducible color triangle. (±96,-96), (±74,-96), (±57,-96), (±44,-96), (±33,-96), (±96,-74), (±74,-74), (±57,-74), (±44,-74).

This enables C_{11}^{I} and C_{11}^{Q} to share 9 information bits.

C_{12}^{I} and C_{21}^{I}: Quantized by 15 levels.

Cutpoints	Representative Value
	± 44
± 39	
	± 34
± 29	
	± 25
± 21	
	± 18

TABLE 3.1. Quantization Table

± 15	± 12
± 10	± 8
± 6	± 4
± 2	0

C_{22}^{I}: Quantized by 9 levels.

Cutpoints	Representative Value
± 26	± 32
± 14	± 20
± 9	± 12
± 3	± 6
	0

C_{22}^{I}, C_{21}^{I}, and C_{22}^{I} have 11 information bits.

TABLE 3.1 - Quantization Table (Continued)

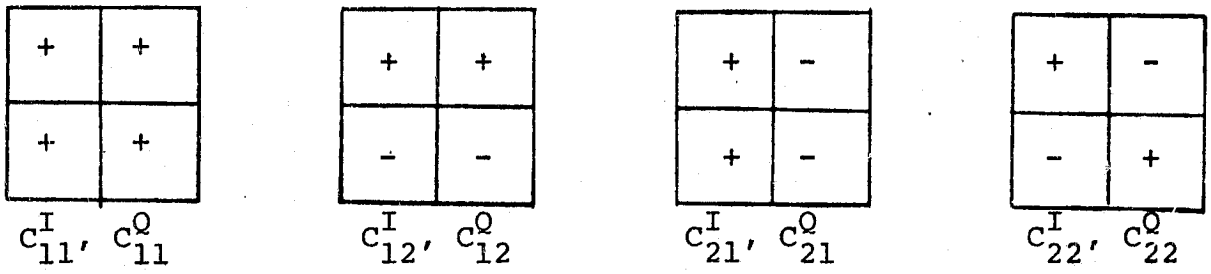


Figure 3.4. Designations of the Chrominance Hadamard Components

more than 9 information bits. However, when both $\hat{I}(t)$ and $\hat{Q}(t)$ have extreme negative values, the resultant chrominance coordinate lies well outside the reproducible color triangle formed with the three primary color vertices. Hence, by combining the representative values and deleting the cutpoint pairs (C_{11}^{I}, C_{11}^{Q}) outside the reporducible color triangle, as shown in Quantization Table 3.1, C_{11}^{I} and C_{11}^{Q} are effectively encoded by 9 information bits.

C_{12}^{I}, C_{21}^{I}, and C_{22}^{I} share 11 information bits. The overall bit requirement for the chrominance components is 20 bits per subpicture. Thus, the resultant two dimensionally compressed color data requires 3.25 bits per picture element (based upon 512 x 480 picture elements per frame), or equivalently, 798.72 Kbits per frame. Therefore, by transmitting the color pictures using only two dimensionally compressed data, the overall bit rate requirement is 23.9616 Megabits per second.

Further bit rate reduction is possible by applying a motion detection method similar to that described in Section 2.2. Here, the object movement or object mutation can be defined as either a change of brightness intensity, or a shift of chrominance coordinates within each defined motion region. This can be done, as in Section 2.2, by comparing C_{11}^{Y}, C_{11}^{I} and C_{11}^{Q} with their predecessors. When motion is detected, the information within the motion region is updated

by transmitting the quantized differences of the d.c. components. Position markers described in Section 2.2 can be applied here directly.

Although a motion detection technique is ideal for minimal data rate transmission, especially when the object scene is relatively inactive, yet, due to the limited environment of a space vehicle where the color video source encoder locates, the motion detection method is not recommended by this report. For the reason that it requires a frame memory for the d.c. components of approximately 276 Kbits, and a rate buffer of approximately 870 Kbits. Whereas, in using the two-dimensional data compression technique given above, these memories are not required. Since the resultant bit rate is only in the order of 24 Megabits per second, which easily satisfies the downlink constraint of 50 Megabits per second, and thus it is recommended from hardware implementation point of view.

The functional block diagram for the color video encoder and decoder is illustrated in Figures 3.5 and 3.6. To minimize possible accumulated decoding errors, due to the usage of DPCM technique, special line group and field synchronizing codewords are used to confine this accumulative effect to within each individual line group.

The decoder outputs, $\hat{Y}(t)$, $\hat{I}(t)$, and $\hat{Q}(t)$ are filtered by low pass networks to insure smooth analog transitions.

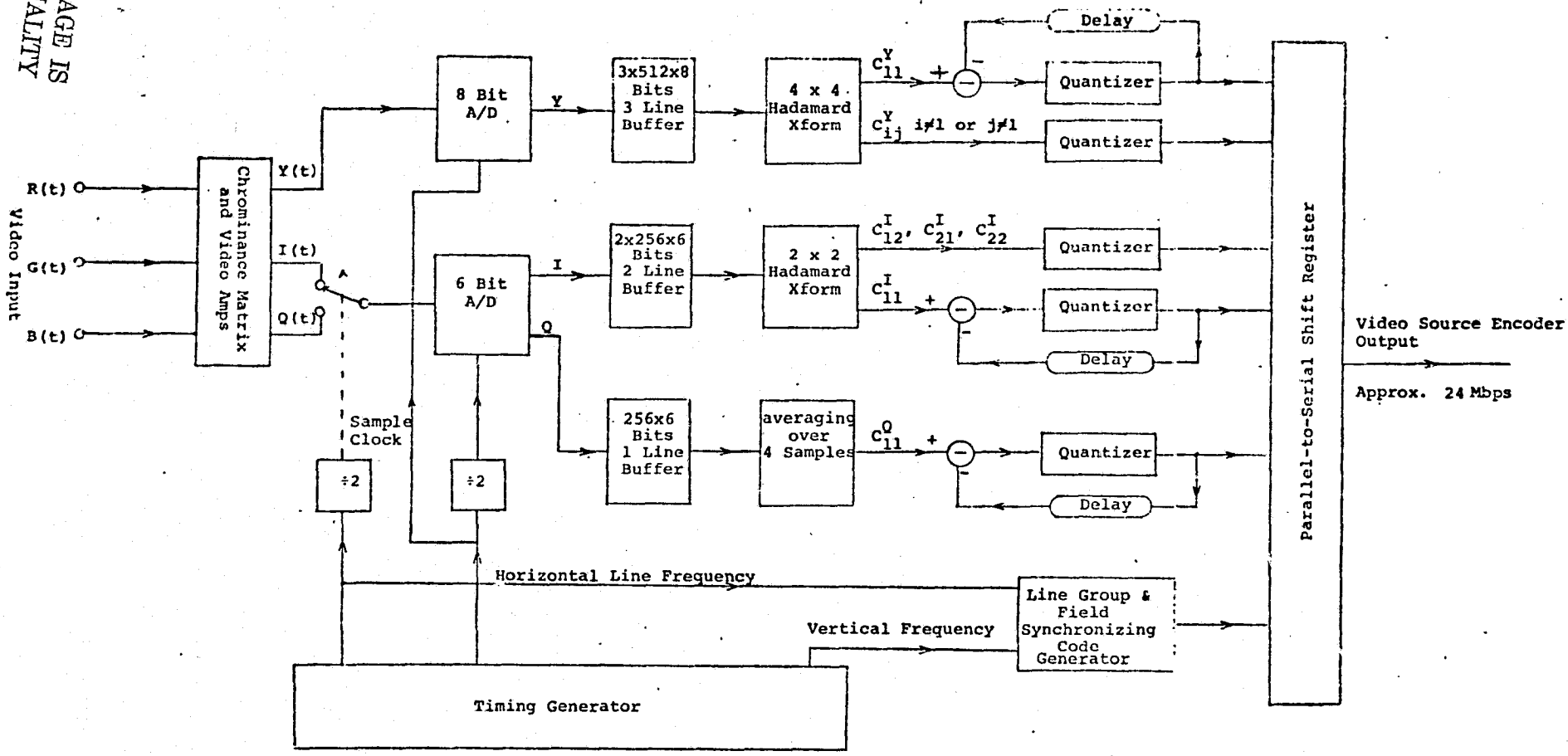


Figure 3.5. COLOR VIDEO SOURCE ENCODER

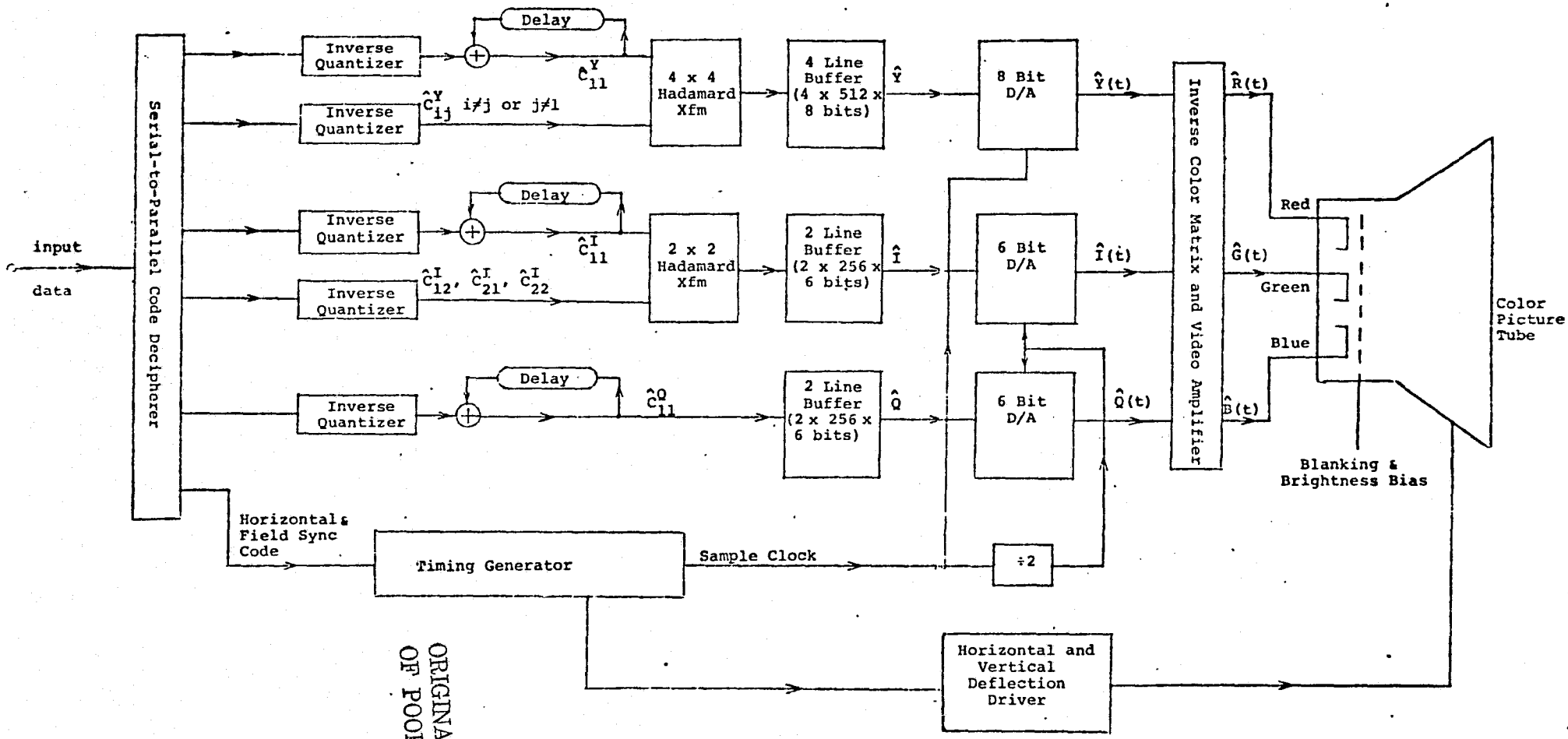


Figure 3.6. COLOR VIDEO SOURCE DECODER

Typically, the cutoff frequencies for Q(t) and I(t) are 1/4 and 1/2, respectively, of that for Y(t). This normally results in undesirable chrominance location errors due to different propagation delays of the filters. As in normal color TV reproduction, linear delay lines are used in series with Y(t) and I(t) to optimize the chromatic effect.

Although experiments for the proposed color source coding technique have not been performed due to the lack of color video equipment, yet we are confident that reasonable color reproduction quality is achievable. This can be substantiated by the following response analysis of the proposed color source coding system.

Since the performance of the component $\hat{Y}(t)$ is identical to the black-and-white monochrome video data compression, its feasibility is verified experimentally. Hence, it is sufficient to analyze the response of $\hat{I}(t)$ and $\hat{Q}(t)$. From the quantizations selected for C_{11}^{I} and C_{11}^{Q}, it is easy to verify the following (in response to a step function of amplitude A):

1) The steady state error for each chrominance component is less than $\frac{1}{64} E_M$, where E_M is the maximum chrominance amplitude (which normally determines the dynamic range of the A-to-D converter).

2) For a step function amplitude, $A \geq \frac{1}{3} E_M$, the decoder can approximate to within 20% of A in one subpicture time, i.e., approximately 500 ns.

3) The steady state (i.e. to within $\pm \frac{1}{64} E_M$) can be reached within 3 subpicture times.

Hence, the equivalent horizontal bandwidth for $\hat{Q}(t)$ is about the 500 KHz that is normally allocated for commercial color TV broadcasting. The vertical resolution is slightly worse due to the 2-to-1 field lacing.

The Quantization Table for C^I_{12}, C^I_{21}, and C^I_{22} was selected to simulate an equivalent horizontal bandwidth of over 1 MHz for $\hat{I}(t)$, with the assumption that the chrominance variation between adjacent picture elements, in the cyan-orange red direction, is relatively small.

4.0 Data Error Rate Analysis

The principal advantage of a digital data communication system is its adaptability to highly efficient discrete channel encoder-decoders, where high output signal-to-noise ratios can be obtained with relatively low transmitted energy. This section presents a data error rate analysis, pertinent to the video source encoder-decoder pairs described in the previous sections. The approach will be to estimate the expected number of picture element errors per frame for the uplink and downlink source and channel coding systems as a function of the energy per picture element-to-single sided noise power denisty ratio, E_{pel}/N_o. The performances of these systems are compared with that of an uncoded (i.e., no source or channel coding) digital video system and it is shown that significant E_{pel}/N_o gains can be achieved with the systems with coding. It should be noted that to obtain good quality video such an uncoded system would require a channel bit rate in excess of that which is presently available.

First we shall develop expressions for the expected number of picture element errors per frame in terms of various error probabilities. Then we shall relate these error probabilities to E_{pel}/N_o and present performance curves for the uplink and downlink source and channel coding systems.

From the point of view of analysis, an exact mathe-

matical model is difficult to prescribe and to analyze for these video source encoder-decoder pairs. Thus, in the sequel a set of mathematical assumptions are made to facilitate the analysis. These are:

1) We shall distinguish three types of source data: the d.c. components, non-d.c. components, and synchronizing codewords. The d.c. components are transmitted differentially either in the two-dimensionally compressed mode or the updating mode. Due to the fact that the accuracy of reconstructed data is completely dependent on the accuracies of previous reconstructed data, this differential method has the inherent disadvantages that data errors can propagate either horizontally, in the two-dimensionally compressed mode, or sequentially (i.e. from frame to frame) in the updating mode. In contrast, the effects of erroneous deciphering of the non-d.c. components are not propagated. In the analysis we shall assume the propagation effect is permanent, i.e., in the two-dimensionally compressed mode, a d.c. component error causes the subsequent portion of the line group to have only erroneous d.c. components, and likewise, in the updating mode, it causes the subsequent corresponding subpictures to have only erroneous d.c. components, (in other words, we do not allow two or more errors to make one correct deciphering), this is illustrated in Figure 4.1.

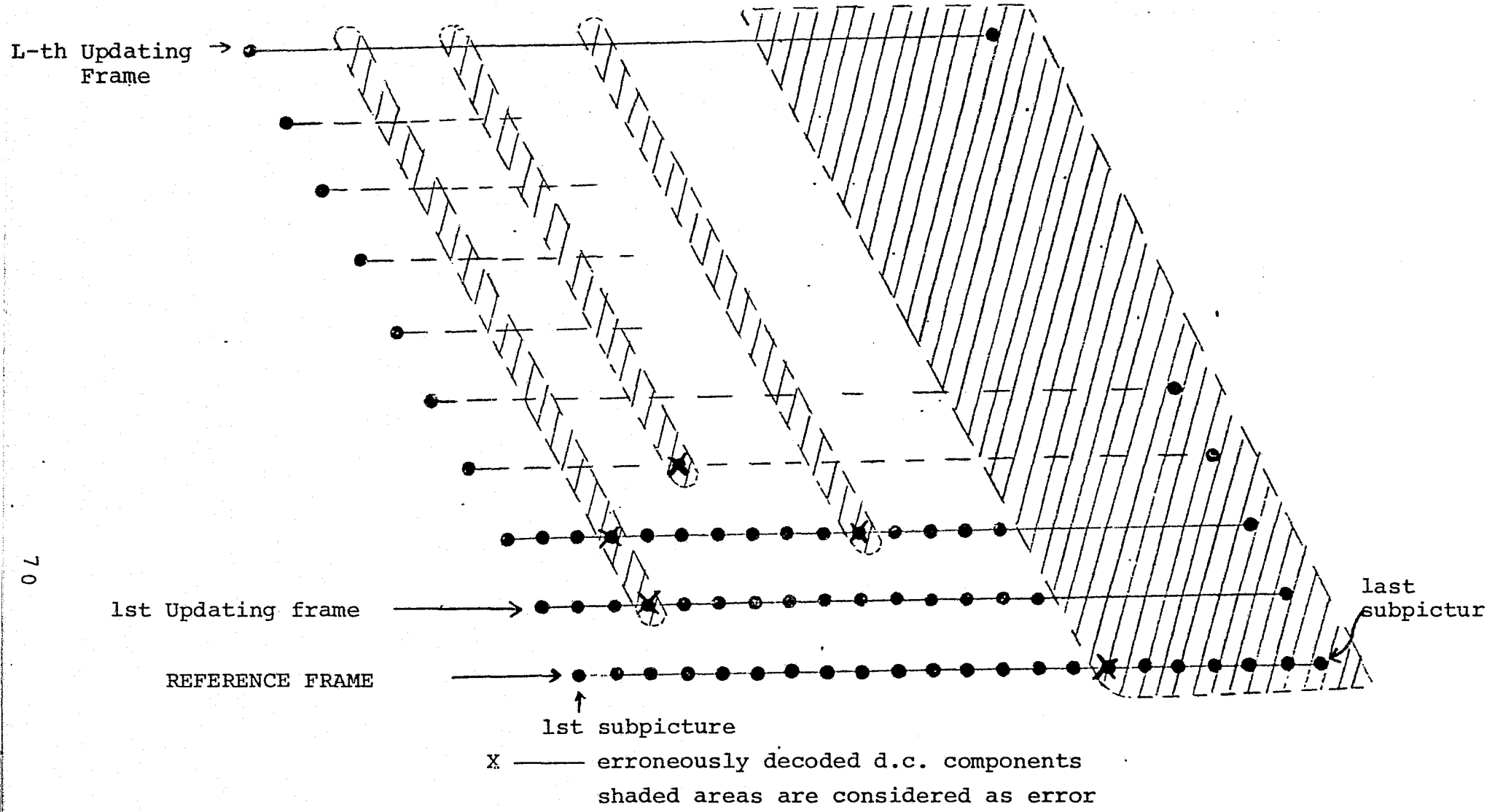


FIGURE 4.1 ERROR PROBAGATION EFFECT OF THE D.C. COMPONENTS OF A TYPICAL LINE GROUP

2) Similarly, if an error occurs in deciphering the synchronizing codeword then the entire line group is considered to be erroneous.

3) The video source encoding alphabets are "ordered" in the sense that a decoding error in some has less effect than others. (For example, if the transmitted d.c. component has code representation -1, then the effect of deciphering as 0 is not as bad as that of deciphering as +15. Generally speaking, distance properties of the channel code could be utilized so that a larger amplitude of decoding error corresponds to lower probability of occurrence). However, for mathematical simplicity, we shall assume that the deciphered components can be either "correct" or "incorrect". This may be interpreted as the probability of deciphering a component such that the error amplitude is beyond tolerance. Furthermore, we shall assume the events of deciphering the source alphabets are independent. With this, let us denote;

p = probability of error in decoding a d.c. component,

α = probability of error in decoding the non-d.c. components of a subpicture, and

γ = probability of error in decoding a synchronizing codeword.

p, α and γ are very small positive real numbers.

4) For the motion detection algorithm, we shall assume that reference frames are transmitted once every L+1 frames (L is a positive integer).

With the above assumptions, we shall divide the error analysis into two parts, one for the two-dimensionally compressed data, and the other for the motion detection algorithm. Our objective is to obtain general mathematical expressions for the expected number of subpicture error per frame in terms of p, α, γ, L and n (where n = number of subpictures per line group = 128). Then the performance of each individual video source encoder-decoder pair can be determined by substituting the appropriate values for p, α, γ, and L.

4.1 General Data Error Expression for the Two-Dimensionally Compressed Video Algorithm

We shall calculate an expression for the expected number of subpicture errors per frame, denoted by the symbol ERF. First, we shall calculate the expected number of subpicture errors per line group, denoted by the symbol, ERL. Then, ERF is related to ERL by:

$$ERF = 120 \cdot ERL \text{ subpictures per frame} \quad \ldots \quad (4.1)$$

Due to the error propagation effect of the d.c. components, it is necessary to calculate the probability that exactly k d.c. components of a line group are correctly decoded, given the synchronizing codeword for the line group is correctly decoded; we shall denote this by P(k).

From the assumptions given above, we have

$$P(k) = pq^k \quad \text{for } 0 \leq k \leq n-1$$

$$\ldots \qquad (4.2),$$

$$= q^n \quad \text{for } k=n$$

where $q=1-p$

(It is not difficult to verify that

$$\sum_{k=0}^{n} P(k) = 1.)$$

Then a line group has exactly k correctly decoded subpictures, (given the synchronizing codeword is correctly decoded), if and only if the line group has k+i correctly decoded d.c. components and exactly k correctly decoded non-d.c. components, (among the subpictures with correct d.c. components), for i=0, 1, 2, ..., n-k.

Therefore,

$Q(k)$ = probability that exactly k subpictures of a line group are correctly decoded given the synchronizing codeword is correct.

$$= \sum_{i=0}^{n-k} \binom{k+i}{i} P(k+i)\ \beta^k \alpha^i \qquad \ldots \qquad (4.3),$$

where $\beta = 1-\alpha$.

We have

$$E\left[\begin{array}{l}\text{number of incorrect sub-}\\ \text{pictures per line group}\end{array}\middle|\begin{array}{l}\text{given the synchronizing}\\ \text{codeword is correct}\end{array}\right]$$

$$= \sum_{k=0}^{n} (n-k)\ Q(k) \qquad \ldots \qquad (4.4)$$

Using the following summation equality:

$$\sum_{k=0}^{n}\sum_{j=0}^{n-k} f(k,j) = \sum_{k=0}^{n}\sum_{j=0}^{k} f(k-j,j),$$

for arbitrary integer function f, we have:

$$\sum_{k=0}^{n} Q(k) = \sum_{k=0}^{n} P(k) = 1$$

and

$$\sum_{k=0}^{n} kQ(k) = \sum_{k=0}^{n}\sum_{i=0}^{n-k} k\binom{k+i}{i} P(k+i)\beta^k\alpha^i$$

$$= \beta \frac{d}{dx}\left\{\sum_{k=0}^{n}\sum_{i=0}^{n-k}\binom{k+i}{i}P(k+i)\ x^k\alpha^i\right\}_{x=\beta}$$

$$= \beta \frac{d}{dx}\left\{\sum_{k=0}^{n} P(k)\ (x+\alpha)^k\right\}_{x=\beta}$$

$$= \beta \sum_{k=0}^{n} kP(k)$$

$$= \beta\left\{p\sum_{k=0}^{n-1} kq^k + nq^n\right\}$$

$$= \beta q(1-q^n)/p$$

Hence, (4.4) is equal to

$$n-\beta q(1-q^n)/p.$$

Therefore,

$$ERL = (1-\gamma)\left\{1-\beta q(1-q^n)/p\right\} + n\gamma$$

$$= n-\beta(1-\gamma)\ q(1-q^n)/p,$$

and

$$ERF = 120\left\{n - \frac{\beta(1-\gamma)\ q(1-q^n)}{p}\right\} \text{ subpictures per frame} \quad \ldots (4.5)$$

4.2 General Data Error Expression for the Motion Detection Algorithm

Likewise, we shall obtain an expression for the expected number of subpicture errors per frame. Since there are L updating frames for each reference frame, it is clear that the expected error gets progressively worse towards the L-th updating frame. Let us denote:

$ERF(\ell)$ = the expected number of subpicture errors per frame for the ℓ-th updating frame

ERL(ℓ) = the expected number of subpicture errors per line group for the ℓ-th updating frame,

For ℓ=0, 1, 2, ..., L,

with ℓ=0 corresponding to the reference frame.

Here the error propagation effect of the d.c. components not only occurs horizontally in the reference frame, but also occurs sequentially in the updating mode. First, we have to calculate the probability that exactly k d.c. components of a line group, at the ℓ-th frame, are correctly decoded, given that the synchronizing codewords up to the ℓ-th frame are correctly decoded. We shall denote this by:

$$P_{\ell}(k) \qquad \begin{matrix} k = 0, 1, 2, \ldots n \\ \ell = 0, 1, 2, \ldots L \end{matrix}$$

Clearly we have

$$\begin{aligned} P_0(k) &= pq^k \quad 0 \leq k \leq n-1 \\ &= q^n \quad k=n \end{aligned} \qquad \ldots \qquad (4.6)$$

(i.e., (4.2)).

Since a line group of the ℓ-th frame has k correctly decoded d.c. components, (given the synchronizing codewords are correct), if and only if its precessor, (i.e., at the ℓ-1st frame) has k+i correctly decoded d.c. components, followed by exactly k correctly decoded d.c. components (among the k+i correctly decoded ones), for i=0, 1, 2, ..., n-k, we have

$$P_{\ell}(k) = \sum_{i=0}^{n-k} \binom{k+i}{i} P_{\ell-1}(k+i) q^k p^i \qquad \ldots \qquad (4.7)$$

The probability that exactly k subpictures of a line group, at the ℓ-th frame, are correctly decoded, given the synchronizing codewords are correct, (as in (4.3)), is given by

$$Q_\ell(k) = \sum_{i=0}^{n-k} \binom{k+i}{i} P_\ell(k+i)\ \beta^k \alpha^i \qquad \ldots \qquad (4.8)$$

Therefore, the expected number of subpicture errors per line group, given the synchronizing codewords are correct, is

$$\Lambda(\ell) = \sum_{k=0}^{n} (n-k)\ Q_\ell(k) \qquad \ldots \qquad (4.9)$$

To obtain a closed form for (4.9), first we have

$$\Lambda(\ell) = \sum_{k=0}^{n} \sum_{i=0}^{n-k} (n-k) \binom{k+i}{i} P_\ell\ (k+i)\ \beta^k \alpha^i$$

Observe that

$$\sum_{k=0}^{n} \sum_{i=0}^{n-k} \binom{k+i}{i} P_\ell\ (k+i)\ \beta^k \alpha^i$$

$$= \sum_{k=0}^{n} P_\ell(k) \sum_{i=0}^{k} \binom{k}{i} \beta^{k-i} \alpha^i$$

$$= \sum_{k=0}^{n} P_\ell(k) = \sum_{k=0}^{n} \sum_{i=0}^{n-k} \binom{k+i}{i} P_{\ell-1}(k+i) q^k p^i$$

$$= \sum_{k=0}^{n} P_{\ell-1}(k) \ (p+q)^k$$

$$= \sum_{k=0}^{n} P_{\ell-1}(k) = \sum_{k=0}^{n} P_0(k) = 1 \ \ldots \quad (4.10),$$

and

$$\sum_{k=0}^{n} \sum_{i=0}^{n-k} k \binom{k+i}{i} P_\ell(k+i) \ \beta^k \alpha^i$$

$$= \beta \frac{d}{dx} \left\{ \sum_{k=0}^{n} \sum_{i=0}^{n-k} \binom{k+i}{i} P_\ell(k+i) x^k \alpha^i \right\}_{x=\beta}$$

$$= \beta \sum_{k=0}^{n} k \ P_\ell(k).$$

Since

$$\sum_{k=0}^{n} k \ P_\ell(k) = \sum_{k=0}^{n} \sum_{i=0}^{n-k} k \binom{k+i}{i} P_{\ell-1}(k+i) \ q^k p^i$$

$$= q \frac{d}{dx} \left\{ \sum_{k=0}^{n} \sum_{i=0}^{n-k} \binom{k+i}{i} P_{\ell-1}(k+i) \ x^k p^i \right\}_{x=q}$$

$$= q \sum_{k=0}^{n} k \ P_{\ell-1}(k)$$

$$= q^{\ell} \sum_{k=0}^{n} k\, P_0(k)$$

$$= q^{\ell+1} (1-q^n) / p,$$

therefore, we have

$$\Lambda(\ell) = n - \frac{\beta q^{\ell+1} (1-q^n)}{p} \quad \ldots \quad (4.11).$$

Consequently,

$$ERL(\ell) = (1-\gamma)^{\ell+1} \Lambda(\ell) + n\left(1-(1-\gamma)^{\ell+1}\right)$$

$$= n - \frac{\beta[q(1-\gamma)]^{\ell+1} (1-q^n)}{p} \quad \ldots \quad (4.12),$$

and

$$ERF(\ell) = 120\left\{ n - \frac{\beta[q(1-\gamma)]^{\ell+1}(1-q^n)}{p}\right\} \ldots \quad (4.13).$$

It should be noted that in the above analyses we have made the assumption that an erroneous deciphering of the line group synchronizing codeword results in only one line group error. In reality, the situation is more complicated. Generally, such an error will cause the line group supposed to be updated to remain unchanged, and, in addition, it also causes an erroneous updating on some other line group, which may or may be an incorrectly decoded line group.

In addition, in the analyses, we have assumed that the probability of decoding error for the d.c. components of the reference frames is the same as that for the d.c. components of the updating frames. Actually, these probabilities of decoding error are different, because, in the updating mode, to decode correctly the components of a subpicture the location marker must be correctly decoded; whereas, this restriction does not exist in the reference mode. In general, if we let,

P_m = probability of motion detected for a subpicture

P_L = error probability of decoding the location marker,

then, we have

p' = probability of error in decoding the d.c. components of the updating frames.

$= P_m [P_L + (1-P_L)p]$.

Using these values, it is not difficult to verify that (4.13) becomes:

$$ERF(\ell) = 120 \left\{ n - \beta[q'(1-\gamma)]^{\ell} q\gamma(1-q^n)/p \right\} \quad \ldots \quad (4.14),$$

where $q = 1-p$

and $q' = 1-p'$.

4.3 Uplink Baseline System Performance

Normally the performance of a communications system with a channel which can be modeled by additive white gaussian noise is specified in terms of the channel coding information bit energy-to-noise ratio, E_b/N_o, required for

a certain error rate. However, when comparing systems with source coding and channel coding the performance should be specified in terms of the received energy-to-noise ratio. Here we will estimate the noisy channel performance of video source coding and channel coding systems by determining the picture element energy-to-noise ratio, E_{pel}/N_o, they require the achieve a certain expected number of picture element errors per frame. Of course, the performance of any system with video compression also depends on the quality of the reconstructed video.

The symbol error probabilities p, p', α, and γ of the previous sections can be upper bounded by the product of the channel coding information bit error probability, P_b, times the number of bits in the symbol. The resulting error probabilities are given in Table 4.1.

When channel coding is used the errors out of the decoder are clustered. However, for small bit error rates and the baseline channel coding systems, the bounds of Table 4.1 represent a very small E_b/N_o increase over the E_b/N_o's obtained using the burst error statistics of the channel decoder.

Thus by using these bounds in (4.13) and assuming that when a subpicture error occurs, all 16 picture elements in the subpicture are in error, we can obtain an

Error Probability	Error Probability Upper Bounds	
	Uplink	Downlink
p	$5\ P_b$	$14\ P_b$
p'	$12\ P_b$	———
α	$27\ P_b$	$38\ P_b$
γ	$7\ P_b$	$14\ P_b$

Table 4.1 Error Probability Bounds

expression for the expected number of picture element errors per frame as a function of the channel coding bit error rate, P_b. For a particular channel coding system, this error rate, and thus the expected number of picture element errors per frame, can be related to system energy-to-noise ratios.

The baseline uplink system is assumed to consist of the following.

Source Coding

- Motion detection algorithm
- 120 x 128 4 x 4 subpictures/frame
- 1/8 bit/black-and-white picture element

Channel Coding

- Constraint length 7
- Rate 1/3
- Information bit rate up to 1 Mbps

Modulation

- BPSK or equivalent

The bit error rate performance of this channel coding system has been determined by computer simulation for error rates greater than 10^{-5} and by Viterbi's transfer function bounding technique [9] for error rates of less than 10^{-5}. Since this bounding technique does not take into account the receiver quantization loss, 0.2 dB has been added to the results obtained with the bounding technique to account

for the loss in using 3-bit receiver quantization. The high and low error rate results are given in Figures 4.2 and 4.3, respectively.

These channel coding results are for a code with subgenerators

$$1111001$$
$$1100101$$
$$1011011 \qquad \ldots \qquad (4.14)$$

This choice of generators achieves a free distance of 14 rather than the maximum possible free distance of 15. However, this code has a smaller bit error rate than the codes with a free distance of 15 for bit error rates of less than about 10^{-11}.

Figure 4.4 gives uplink performance curves obtained using the channel coding results of Figures 4.2 and 4.3, and the relationship

$$\frac{E_{pel}}{N_o} = \frac{1}{8} \frac{E_b}{N_o} \qquad \ldots \qquad (4.15)$$

As a basis for determining the E_{pel}/N_o gain achieved with this source and channel coding scheme, let us determine the expected number of picture element errors versus E_{pel}/N_o performance of a system with no source or channel coding. Assume 7 bits per picture element. Then this system would require a data rate of 51.6 Mbps which considerably exceeds

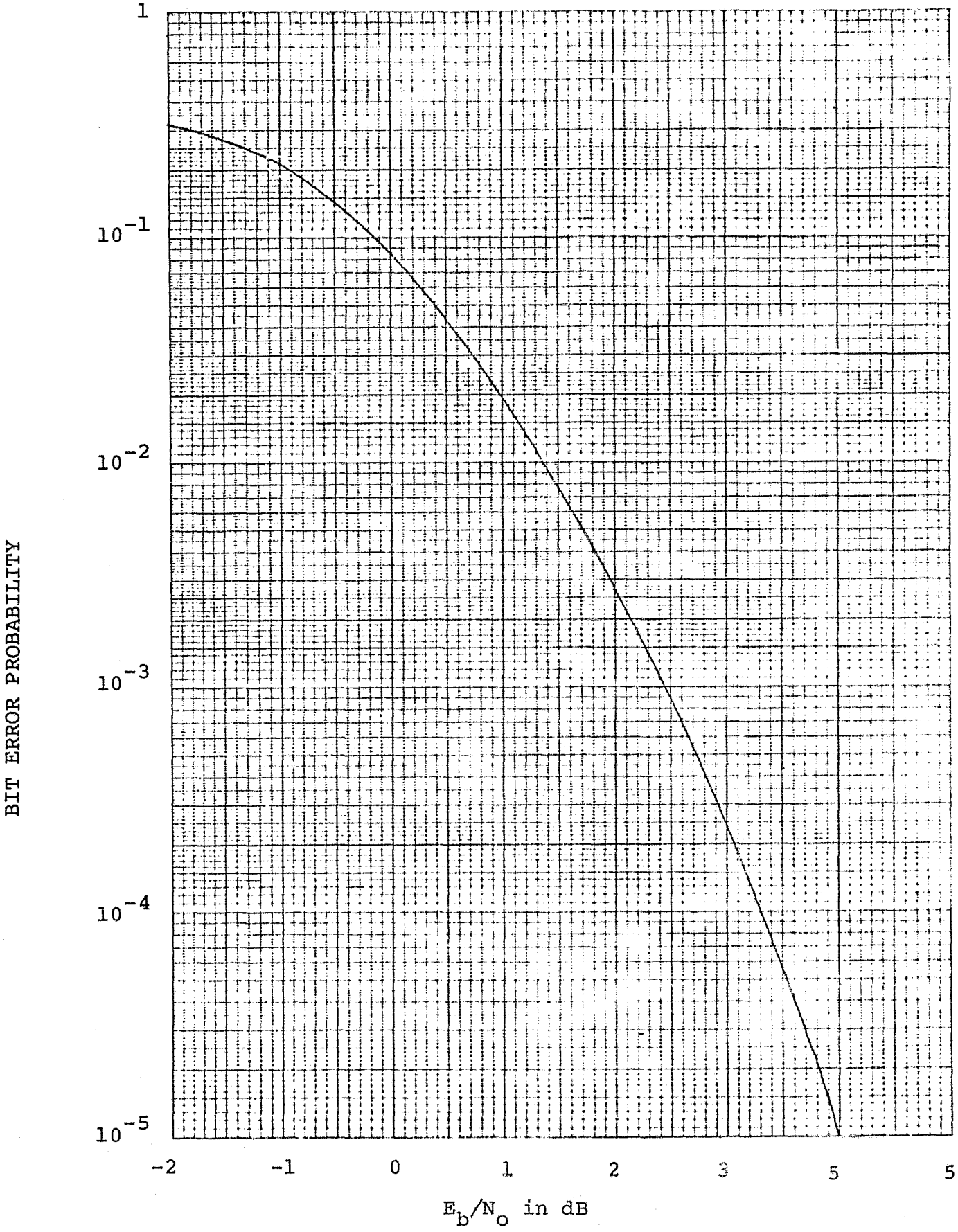


Figure 4.2 Bit error probability performance of a K=7, R=1/3 convolutional coding system obtained by simulation.

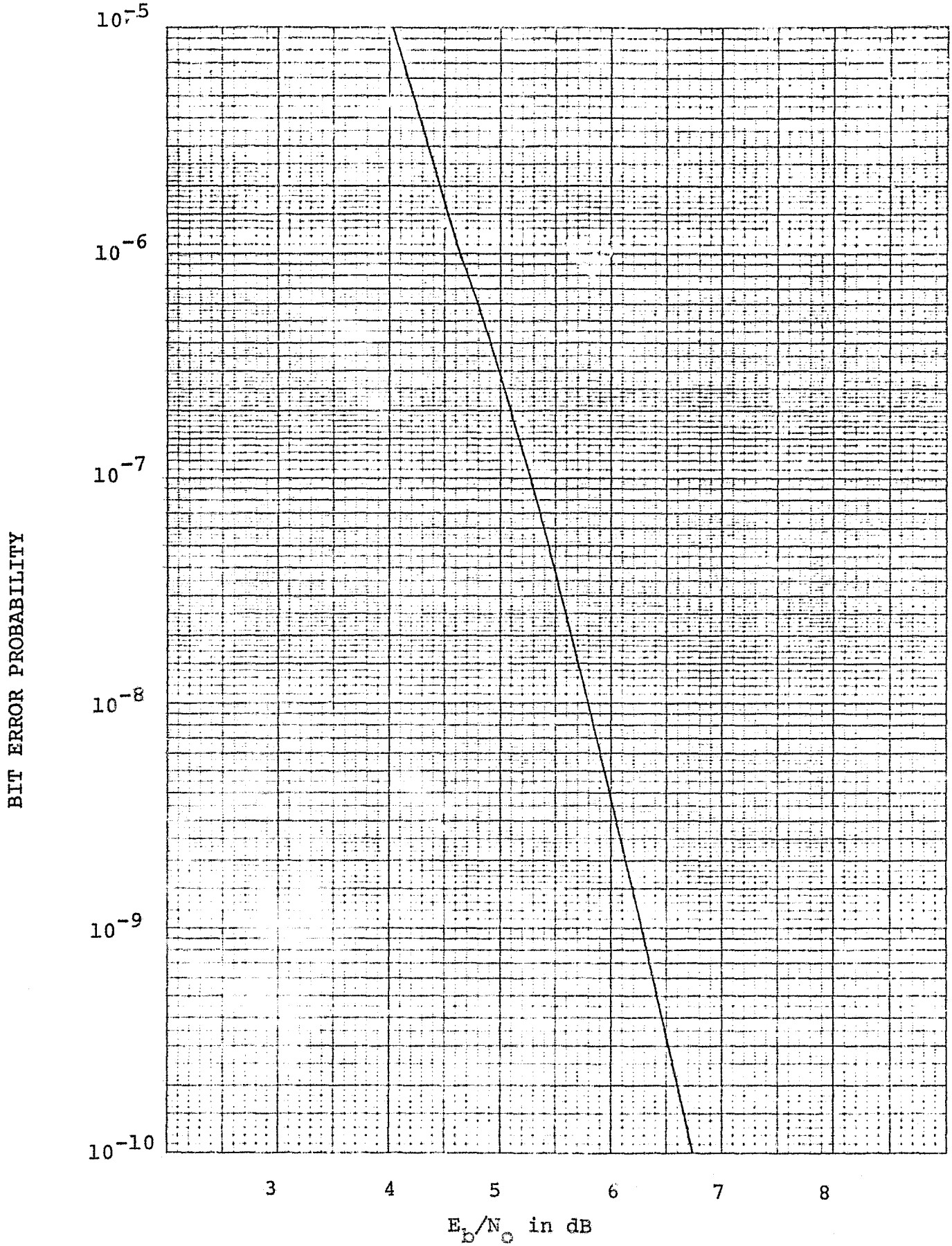


Figure 4.3 Small bit error probability performance of a K=7, R=1/3 convolutional coding system obtained from bounds

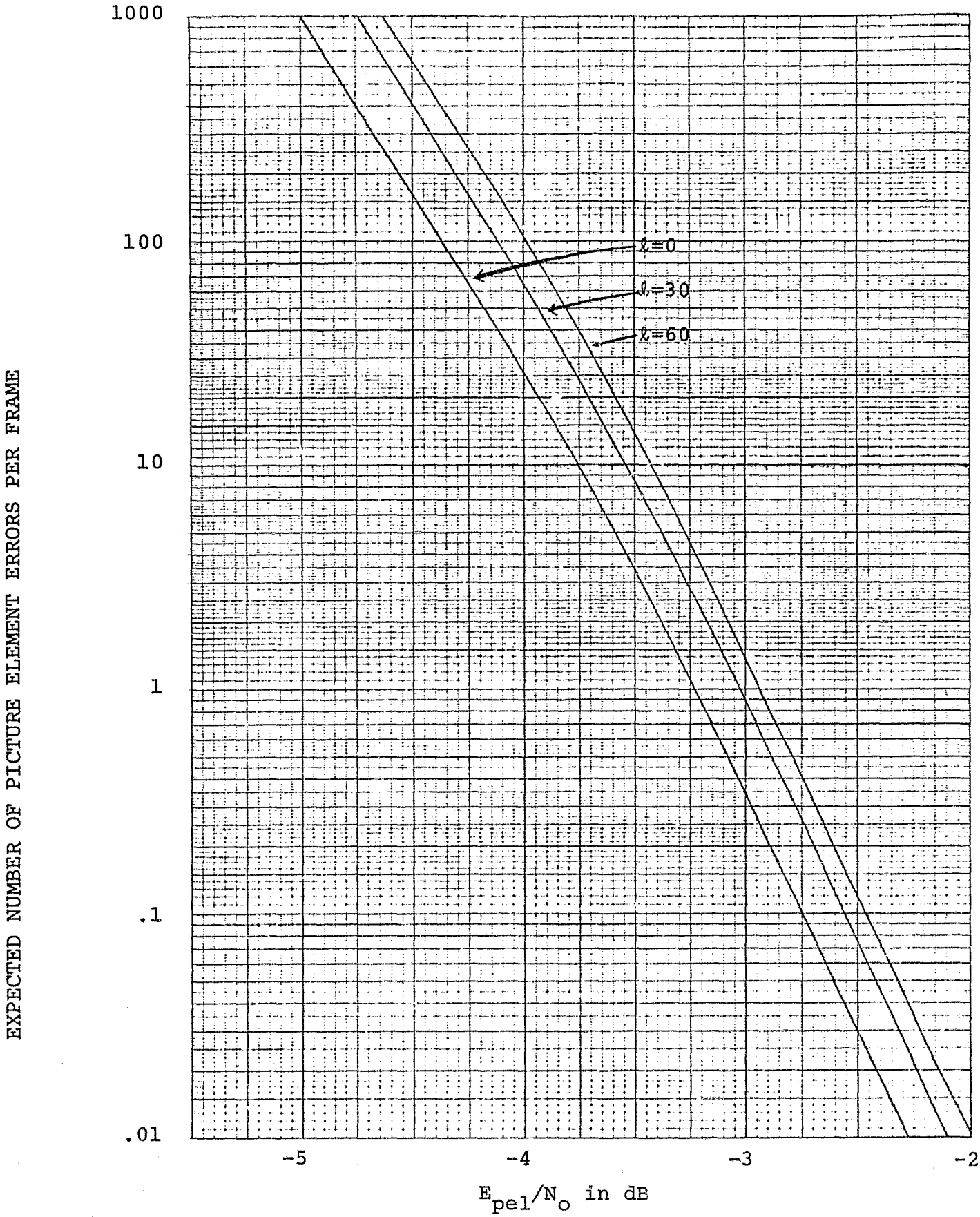


Figure 4.4 Noisy channel performance of the uplink baseline source and channel coding system.

the 3 Mbps channel bit rate available without channel coding. Thus some type of source coding would be necessary.

With the same assumptions as for the baseline system, the expected number of picture element errors per frame with this uncoded system is

$$X_{uncoded} = 16(128)(120)(7P_b)$$

$$= 1720320\ P_b \qquad \ldots \qquad (4.16)$$

where

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \qquad \ldots \qquad (4.17)$$

and

$$Q(y) = \int_y^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \qquad \ldots \qquad (4.18)$$

The picture element energy-to-noise ratio is related to the channel symbol energy-to-noise ratio by

$$\frac{E_{pel}}{N_o} = 7\,\frac{E_b}{N_o} \qquad \ldots \qquad (4.19)$$

Figure 4.5 gives the performance of this uncoded system obtained from equotions 4.16 thru 4.19. In terms of this performance measure, this uncoded system is seen to be significantly inferior to that of the coded system.

4.4 Downlink Baseline System Performance

The noisy channel performance of the downlink source and channel coding system can be estimated using the same

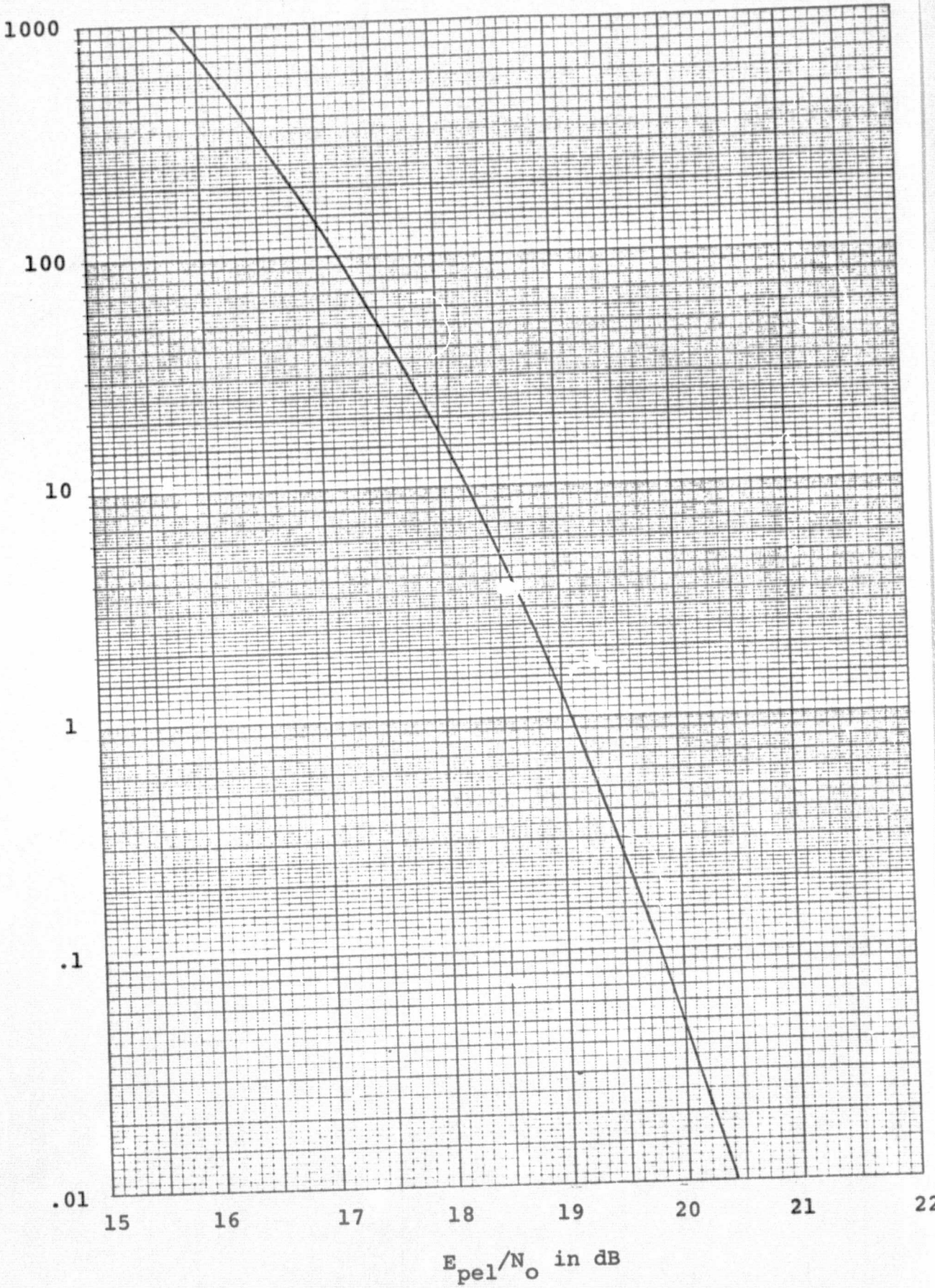


Figure 4.5 Noisy channel performance of an uncoded uplink digital video communications system with 7 bits/picture element.

C.2

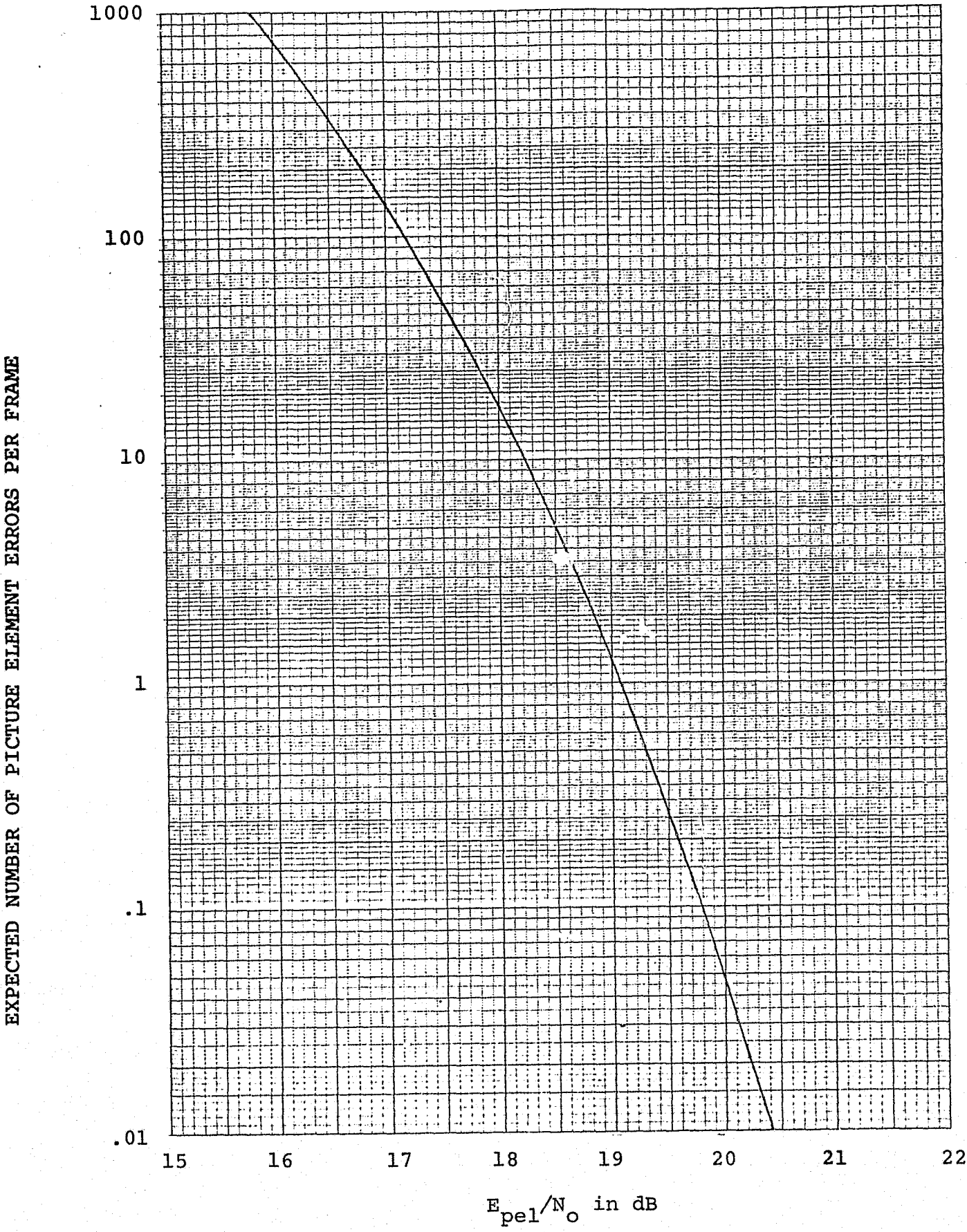


Figure 4.5 Noisy channel performance of an uncoded uplink digital video communications system with 7 bits/picture element.

C. 2

procedure as for the uplink. In this case the baseline system is assumed to consist of the following.

Source Coding

- Two-dimensional algorithm (i.e., ℓ=0 only)
- 120 x 128 4 x 4 subpictures/frame
- 3.25 bits/color picture element

Channel Coding

- Constraint length 7
- Rate 1/2
- Information bit rate up to 50 Mbps. This would require parallel decoders for real time operation.

Modulation

- BPSK or equivalent

The choice of 3.25 bits per picture element is believed to be sufficient for the source coding algorithm to produce good quality color video. However, this choice only produces a channel encoder information bit rate of 24 Mbps. Thus a 50% reduction in the channel symbol rate with a doubling of the channel symbol energy is possible. Another possibility is to concatenate an outer channel coding operation to the baseline channel coding scheme. This concatenated coding possibility is discussed in the next section.

Figures 4.6 and 4.7 show the bit error rate performance of the downlink baseline channel coding system and Figure 4.8 gives the noisy channel performance of the downlink source and channel coding system.

Figure 4.9 gives the expected number of picture element errors per frame performance of an uncoded digital color communications system with 13 bits per picture element. As with the uplink case, the uncoded system exceeds the maximum baseline channel symbol rate. Its performance is included for comparison purposes only.

4.5 Concatenated Channel Coding Performance Improvements

The small bit error probabilities reflected in the performance curves of the baseline coding systems indicate that concatenated channel coding may be desirable. The purpose of concatenated coding is either to obtain a small error rate with an overall encoder/decoder implementation complexity which is less than that which would be required by a single coding operation or to improve the error rate performance of an existing channel coding system. Typically the inner code corrects most of the channel errors and then a rather simple, high rate, outer code reduces the error rate to the desired value. Also interleaving is usually required between the coding operations to breakup the error bursts out of the inner coding operation.

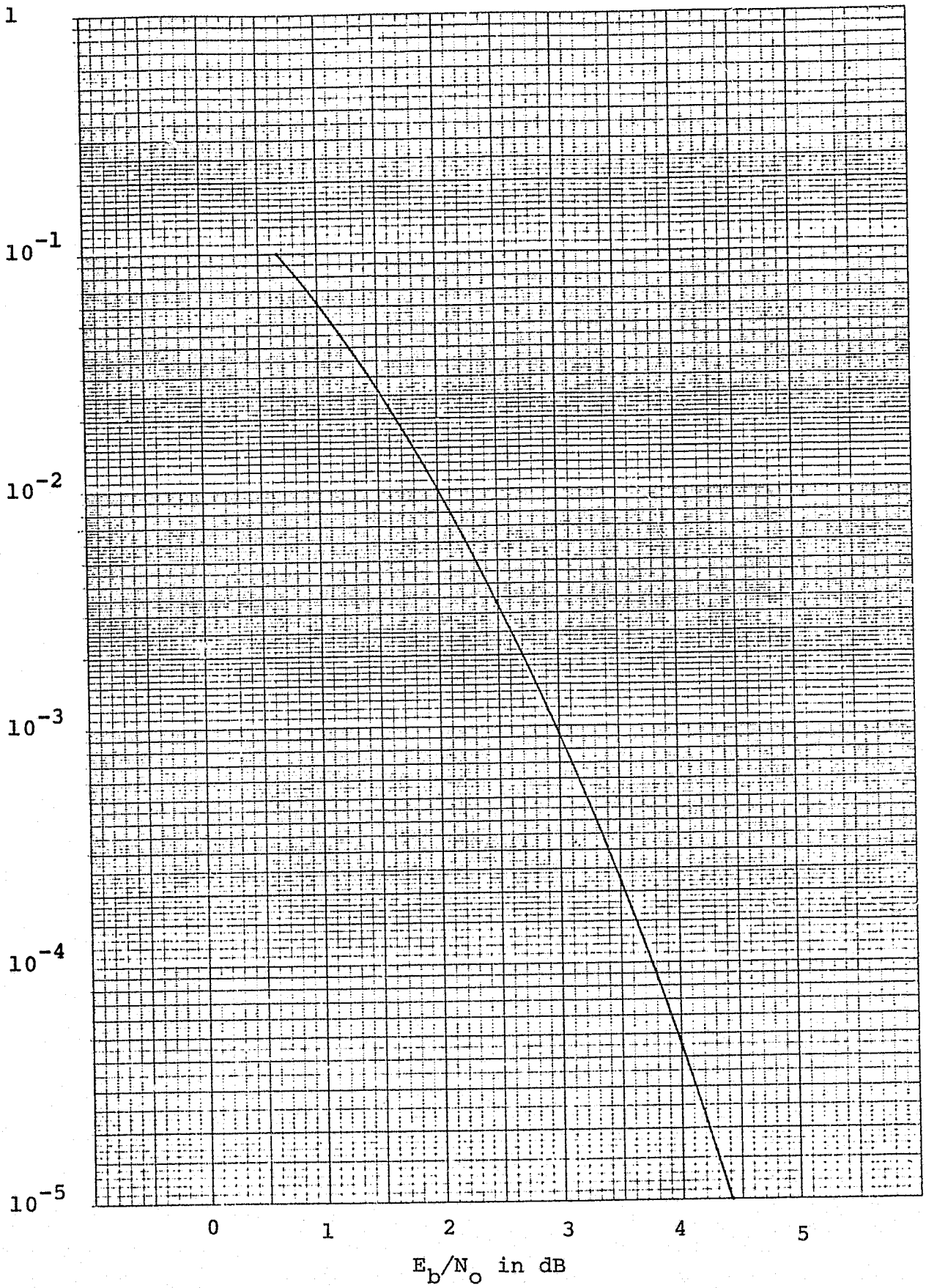


Figure 4.6 Bit error probability performance of a K=7, R=1/2 convolutional coding system obtained by simulation.

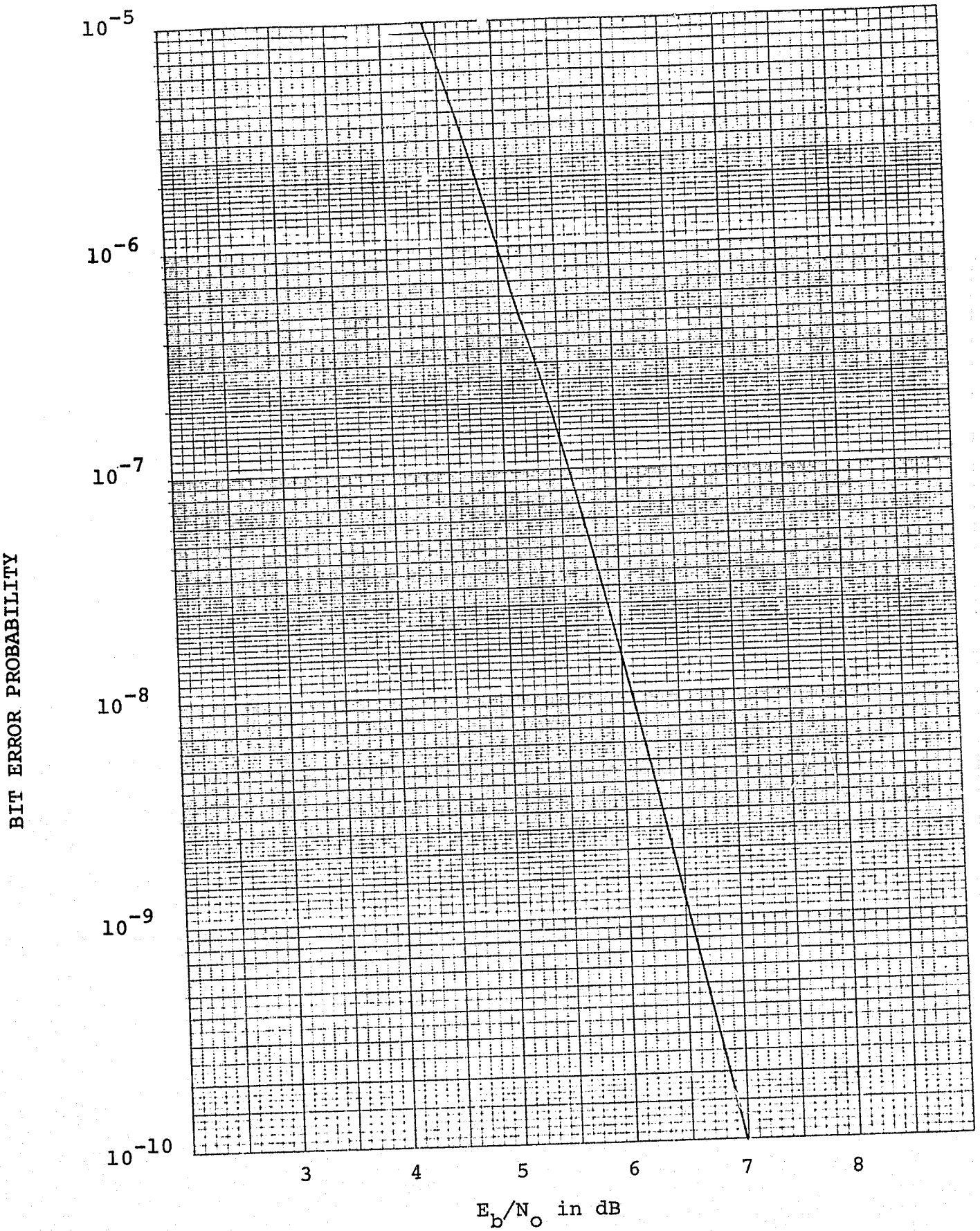


Figure 4.7 Small bit error probability performance of a K=7, R=1/2 convolutional coding system obtained from bounds

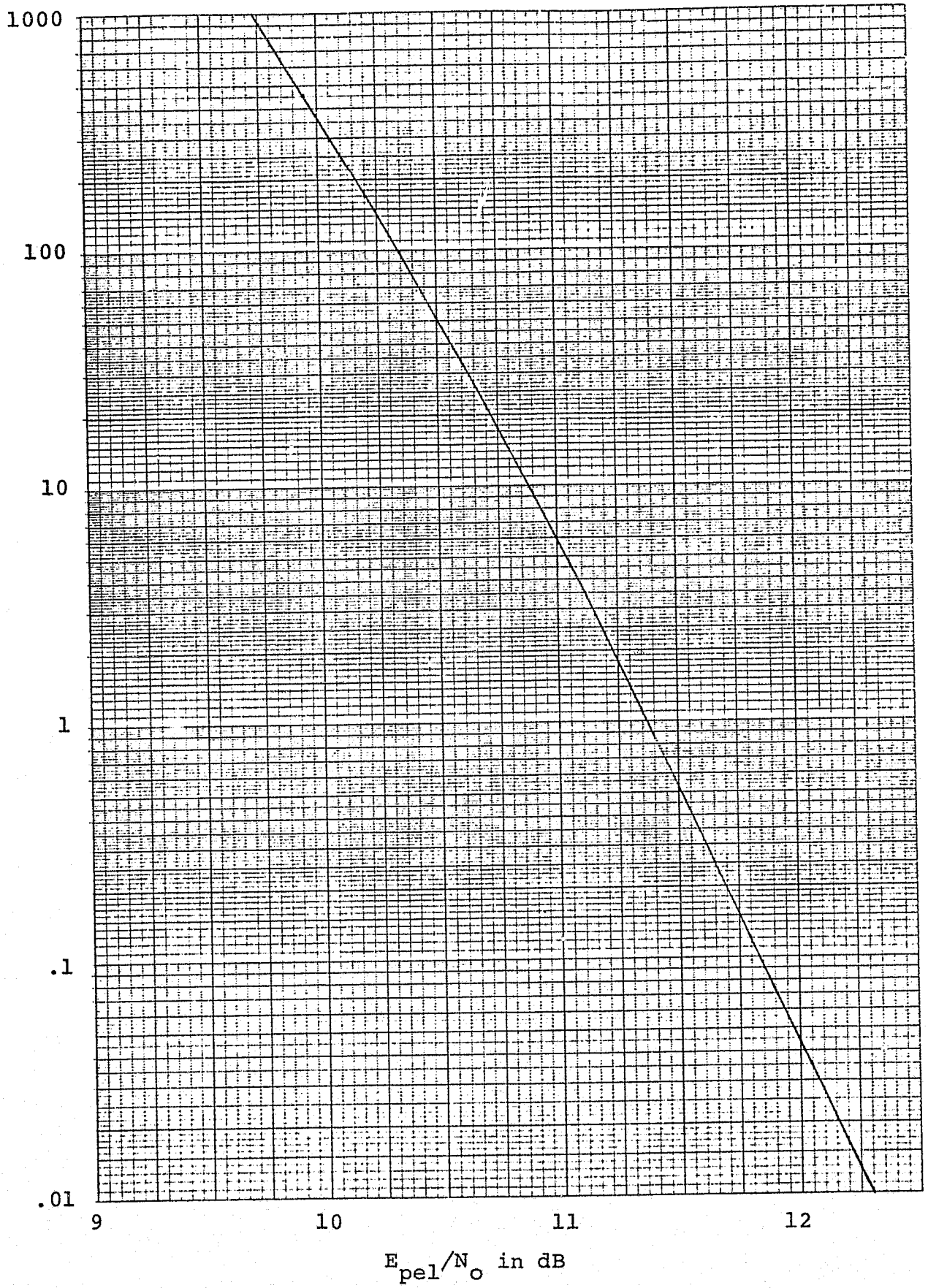


Figure 4.8 Noisy channel performance of the downlink baseline source and channel coding system

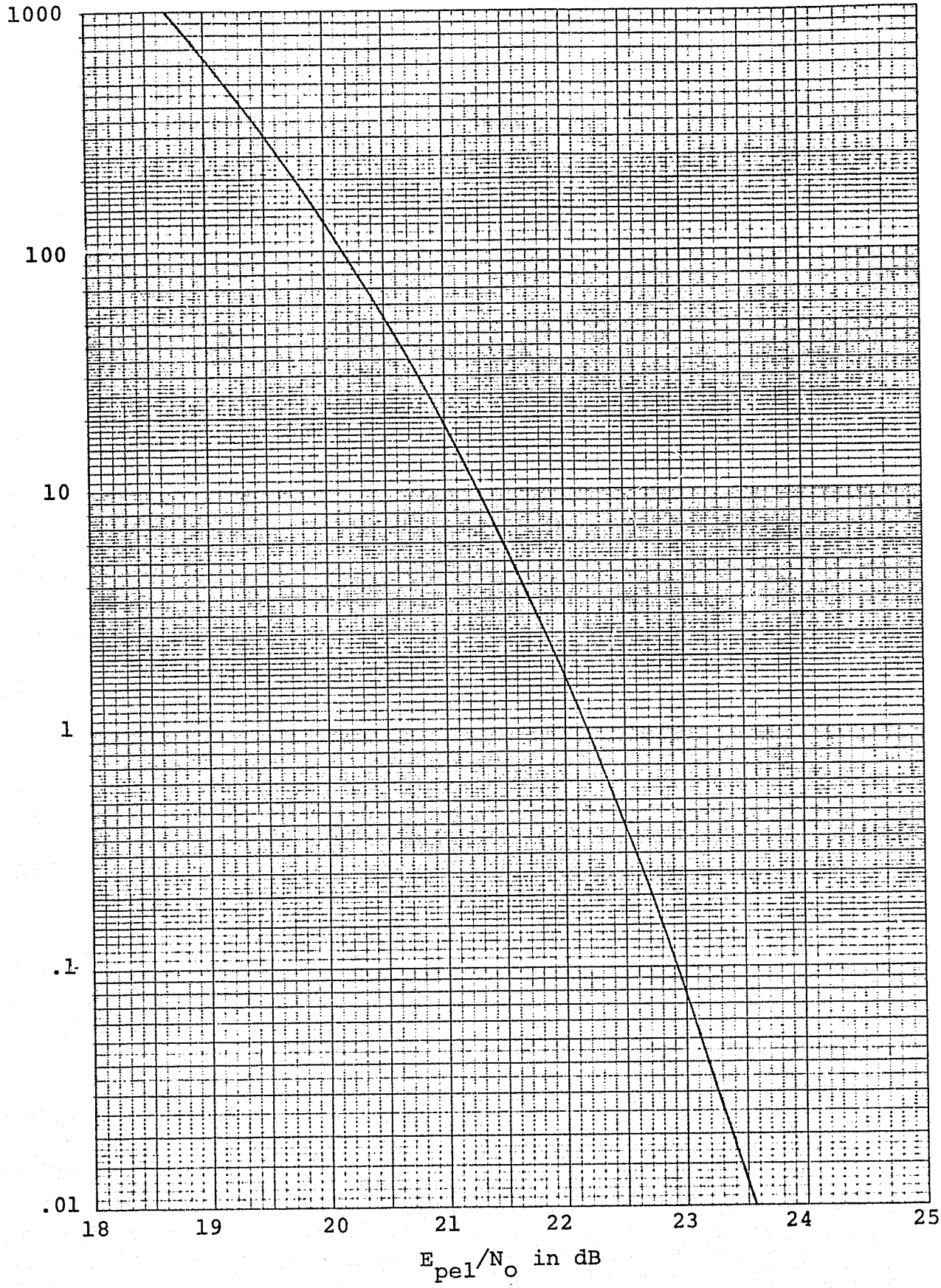


Figure 4.9 Noisy channel performance of an uncoded downlink digital video communications system with 13 bits/color picture element.

4.5.1 Concatenated Channel Coding for All the Data

Concatenanted coding can be used for all the data or just the data with more severe reliability requirements. First we consider the case where concatenated coding is used for all the data. The LINKABIT LF89 feedback-decoded convolutional encoder/decoder is a good choice for the outer coding in such a system. This encoder/decoder has a constraint length 8 rate 3/4 convolutional encoder and a simple feedback decoder which could operate on the hard decision outputs of the inner Viterbi-decoded convolutional coding system. The LF89 is also available with internal interleaving/deinterleaving.

Figure 4.10 compares the bit error rate performance of the baseline uplink coding system with that of a concatenated coding system using the LF89 coding as an outer code and the baseline K=7, R=1/3 convolutional coding as the inner code. In the error probability range of Figure 4.10, the performance of the LF89 coding can be approximated by

$$\text{bit } P_E \simeq 2000\ P_{in}^{3} \qquad \ldots \qquad (4.20)$$

where P_{in} is the bit error rate into the LF89 decoder (i.e., out of the Viterbi decoder). The E_b/N_o is also adjusted to account for the information bit energy change caused by the lower rate of the concatenated coding system.

Figure 4.10 shows that this concatenated coding system can provide significant energy-to-noise ratio gains

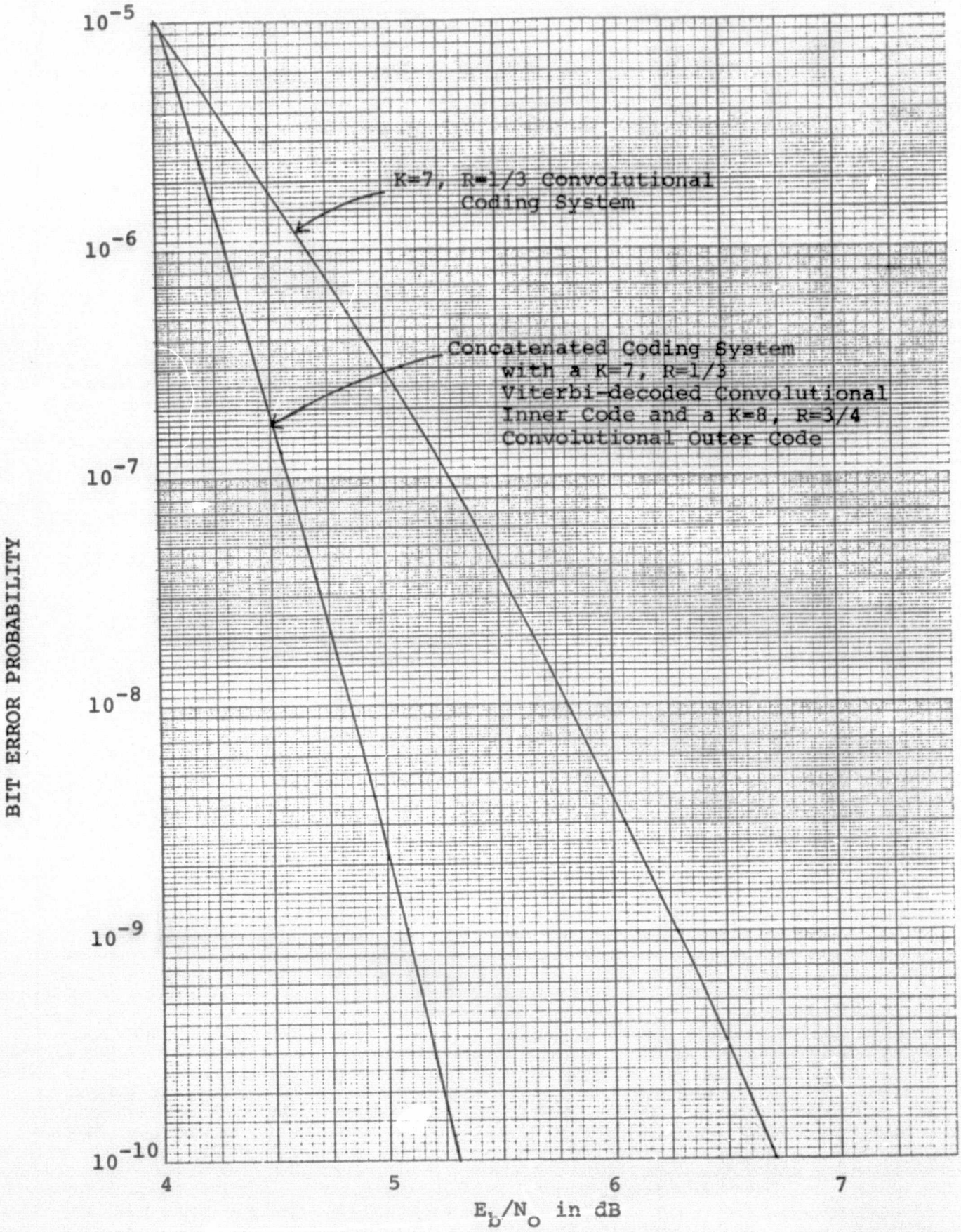


Figure 4.10 Uplink concatenated coding bit error rate performance

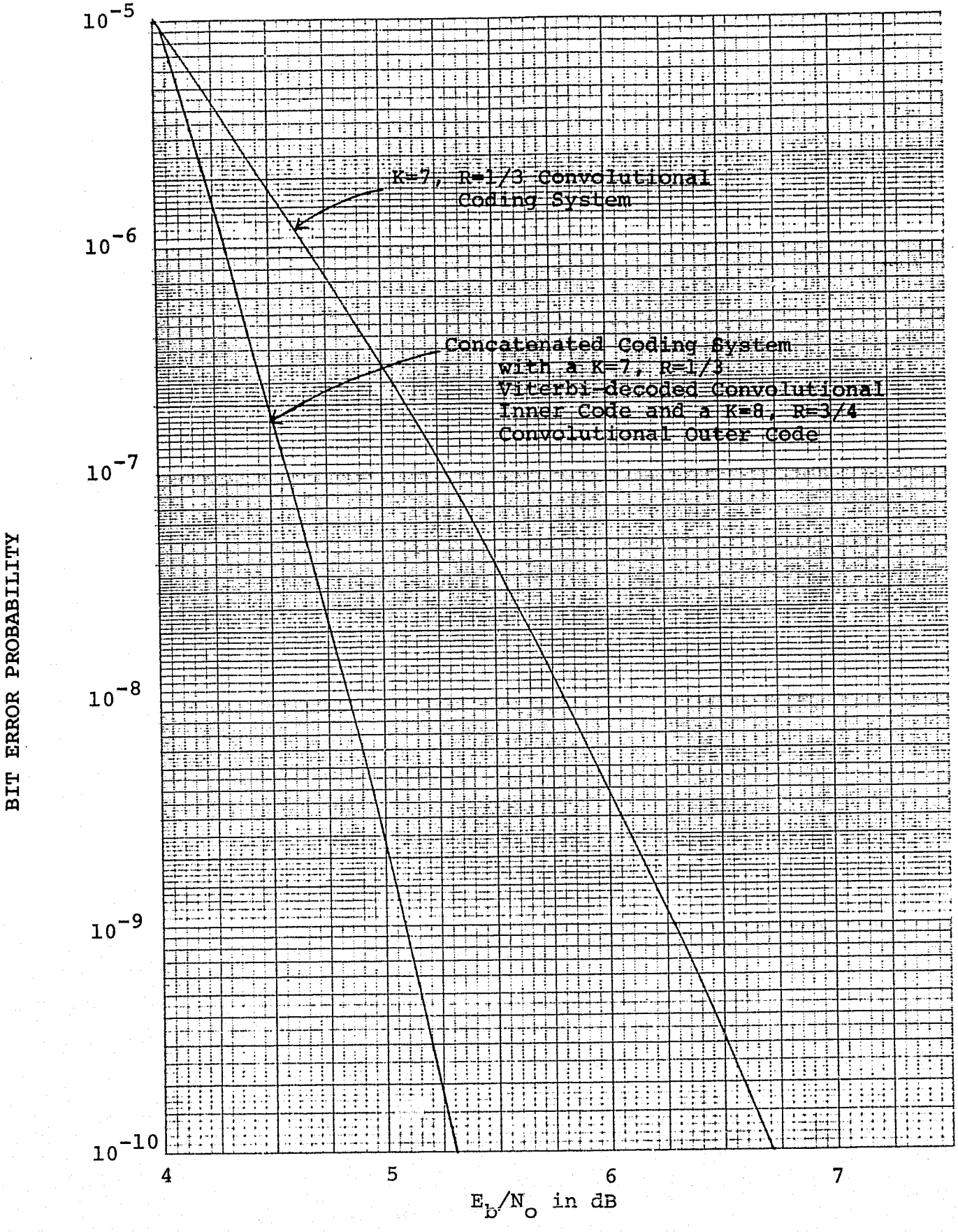


Figure 4.10 Uplink concatenated coding bit error rate performance

when small bit error rates are required. For reference, a 10^{-10} bit error rate produces an expected number of picture element errors per frame of about .009 for the uplink system with $\ell=0$.

To accomodate this concatenated coding scheme on the uplink, it would be necessary to reduce the number of bits per picture element produced by the source encoder by the factor .75 to maintain the same approximate 1 Mbps data rate into the inner encoder. This will degrade the picture quality, especially when motion occurs. However, if the picture quality is still acceptable the additional source compression will provide an E_{pel}/N_o gain of 1.25 dB. Thus the total gain is the dB gain shown in Figure 4.10 plus 1.25 dB. At an expected number of picture element errors per frame of 0.01, this total E_{pel}/N_o gain is 2.65 dB.

Concatenated coding can also be used on the downlink. Performance improvements similar to those of Figure 4.10 can be obtained with the downlink K=7, R=1/2 baseline system as the inner code and the LF89 (or several in parallel to achieve the high data rates) as the outer code. On the downlink the source coding can remain unchanged since the inner code can operate up to 50 Mbps. Thus the E_{pel}/N_o gain is equal to the E_b/N_o channel coding gain achieved by the concatenated coding system over the baseline system.

4.5.2 Concatenated Coding for Only the Reference Symbols

When a small fraction of the data has more severe reliability requirements than the other data, the overall system performance can sometimes be improved by concatenating an outer code to the channel coding system for only the higher reliability data. The advantages of using concatenated coding on only a small fraction of the data rather than on all the data are that the channel symbol rate increase is much smaller and the power of the outer code can be used where it is most needed.

In the uplink video communications system described here an error in a reference symbol is more serious than an error in an updating symbol. So as an example of this type of concatenated coding, let the concatenated coding be used on only the reference symbols. For the uplink system these reference symbols represent approximately 15% of the data presented to the channel coding system.

When the outer coding is an LF89 feedback-decoded convolutional coding system the expected number of picture element errors per frame can be related to the bit error rate of the baseline channel coding system by replacing the updating symbol error probability approximation of Table 4.1 by

$$p = 5\ (2000\ P_b^{\ 3})$$

The 5% average symbol rate increase required by this system can be accommodated by the baseline system without any changes in the video compression ratio. Also this small symbol rate increase only reduces the inner code E_b/N_o by 0.2 dB.

The expected number of picture element errors per frame of this concatenated system is about 0.1 of that obtained with the baseline uplink system (see Figure 4.8). This performance improvement corresponds to an E_{pel}/N_o gain of about 0.5 dB. Subtracting the rate reduction loss, produces a net gain of only 0.3 dB. So this concatenated system requires a larger E_{pel}/N_o than the concatenated system of the previous section. Of course, this concatenated system does not require a larger source compression ratio. Thus this system may be preferable to the previous system when picture quality is considered.

5.0 Computer Simulation

A Fortran program is written to simulate the motion detection technique. The simulation was performed on the LIM Video Controller, whose functional block diagram is illustrated in Figure 5.1. To process a sequence of pictures, the signal is first recorded on the video disc recorder, Ampex model DR-10, (up to 300 frames of video signal, or equivalent to 10 seconds of realtime video signal, can be processed), then the LIM Video Controller processes the signal by extracting 4 consecutive horizontal lines at a time from the video disc recorder. It A-to-D converts the analog signal, and transports the digitized data to the Meta IV CPU for data manipulation. The processed data are then returned to the LIM Video Controller, D-to-A converted back into video signal, and recorded on the video disc recorder. Due to the inherently slow mechanical movements of the record/playback heads, the reference signal and the processed signal are recorded at alternate tracks and by alternate record/playback heads. For example, the odd numbered reference frames and the even numbered processed frames are recorded by record/playback head A, (which controls the top side of the disc), while the even numbered reference frames and the odd numbered processed frames are

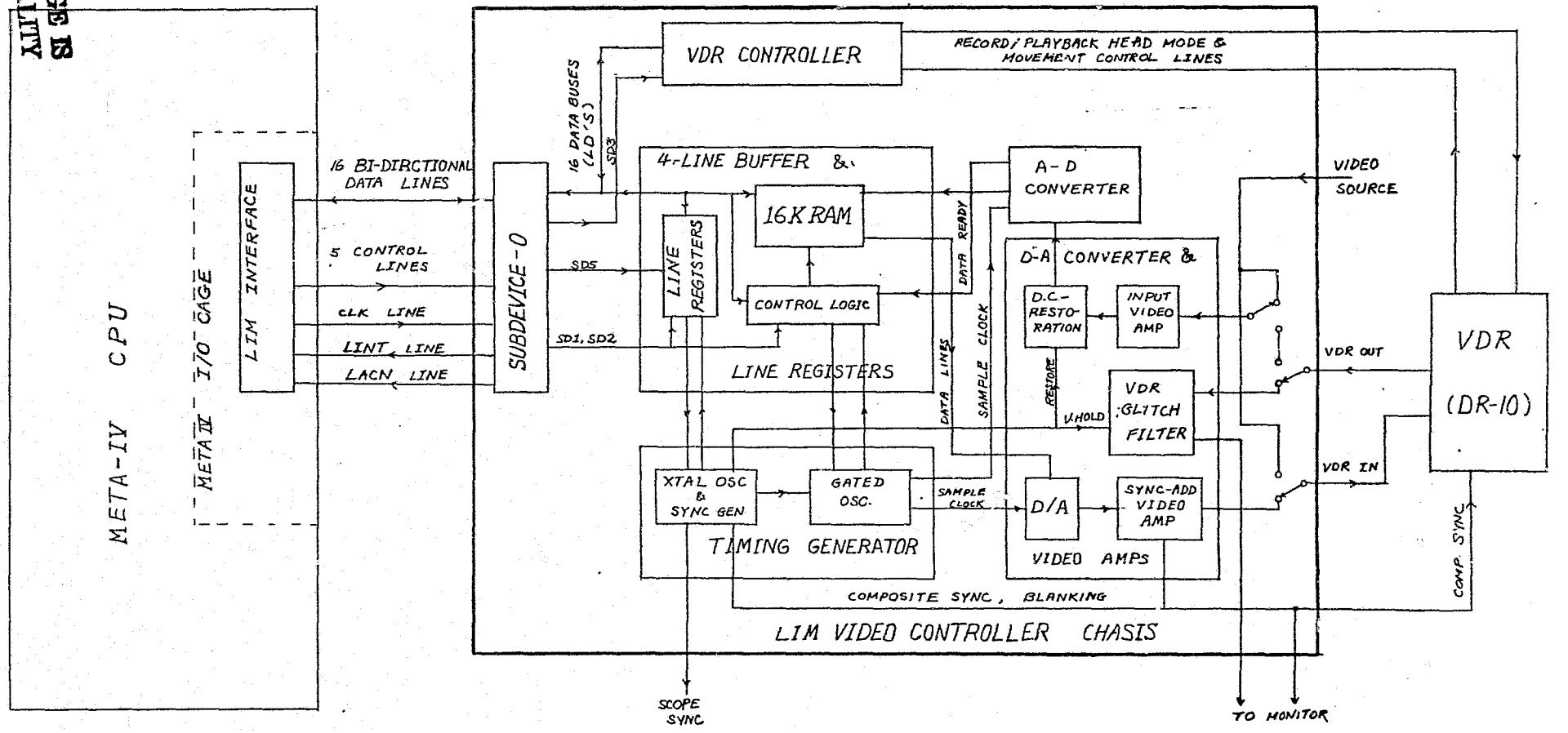
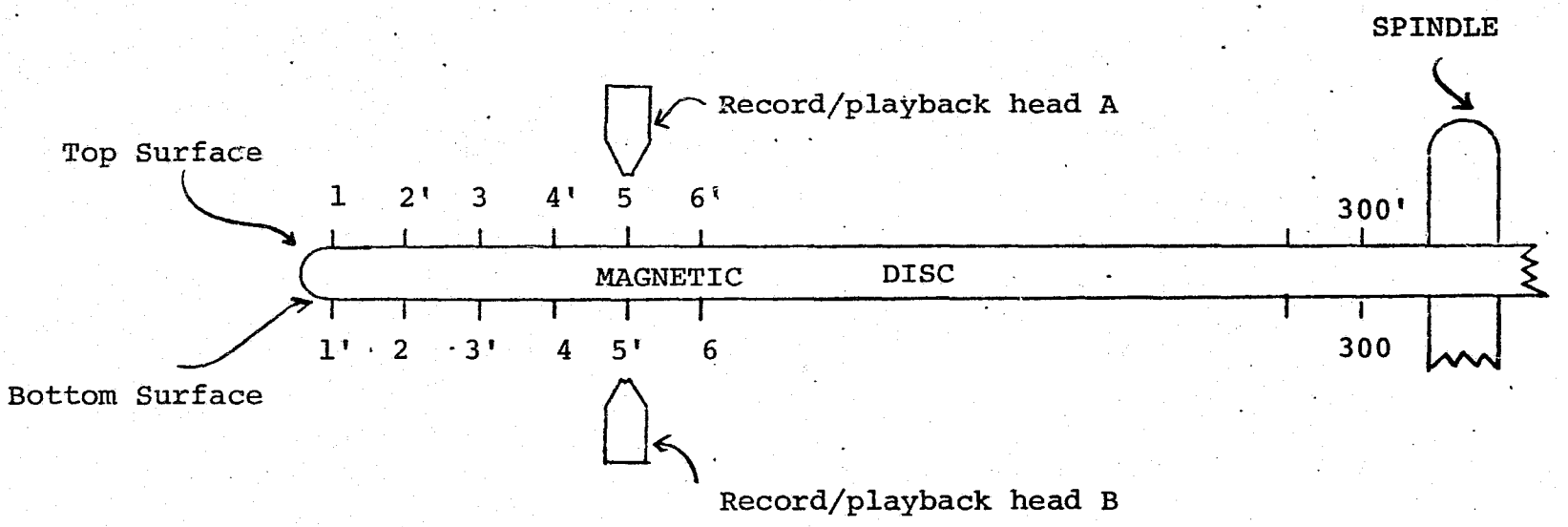


Figure 5.1: LIM VIDEO CONTROLLER FUNCTIONAL BLOCK DIAGRAM

recorded by record/playback head B, (which controls the bottom side of the disc); this is illustrated in Figure 5.2. Due to the fact that the record/playback heads are not perfectly matched, signal variations are resulted in terms of different signal gains and nonuniform intensity changes between heads. Since the motion detection algorithm uses a threshold comparison method, which is sensitive to the stability drifts, therefore, these together with other signal variations resulted from the long term drifts of A-to-D and D-to-A converters, video amplifiers, and timing generator (these intrinsic hardware drifts are very small, yet, due to the long processing time between frames their effects are not negligible in simulating motion detection) should be taken into account for a more accurate interpretation of the effectiveness of the algorithm. The above signal variations have resulted excessive motion regions detected and some unwanted noises, especially when the threshold is set to a high level. To obtain a closer estimation on the number of detected motion regions, another Fortran program is written where the threshold comparisons are made between corresponding motion regions of the adjacent fields. By doing so, the field 2 motion detection is obtained by comparisons of video signals



1, 2, 3, ..., etc: input reference frame number

1', 2', 3', ..., etc: processed frame number

Figure 5.2: Typical Geometrical Orientation of Reference and Processed Frames

of two fields that are obtained by the same record/playback head, while that of field 1 is obtained by different record/playback heads. Our experiments have shown that the number of motion regions detected in field 1 is, in general, substantially higher than that of field 2. This substantiates our claim on the effect of signal variations due to different record/playback heads. However, it should be emphasized that the field-to-field motion detection algorithm mentioned above is not desirable, from quality point of view, for the reason that this method induces extra detected motion regions due to the location errors of motion regions between adjacent fields. In particular, if the scene contains horizontal objects the algorithm causes certain oscillations about the horizontal edges, because here the algorithm attempts to correct the errors due to unmatched locations. The sole purpose of the field-to-field experiment is to obtain a closer estimate on the number of motion regions detected.

The Fortran program for the (Frame-to-Frame) motion detection is given in the following pages. Most of the Fortran subroutines are described in the LIM Video Controller Operator's Manual [6] or in "Study of Efficient Video Compression Algorithm for Space Shuttle Application",

```
// JOB

LOG DRIVE   CART SPEC   CART AVAIL   PHY DRIVE
  0000        0100         0100         0001
                           00A2         0002
                           0CCB         0003
                           CCCC         0004

V2 M08   ACTUAL 16K  CONFIG 16K

// FOR
*LIST ALL
*ONE WORD INTEGERS
*IOCS(1403 PRINTER,2501 READER,DISK,TYPEWRITER,KEYBOARD)
      INTEGER JD1(4162),JD2(4162),BF1(2049,2),BF2(2049,2)
      INTEGER GL2(32),IGL2(32),GN(15,4),IGN(15,4),NGX(15)
      INTEGER CHAR(16,16),DIGIT(3),FNUM(2),ECF,EOF
      INTEGER MPT(129)
      EQUIVALENCE (JD1(1),BF1(1,1)),(JD2(1),BF2(1,1))
      DEFINE FILE 1(780,320,U,K)
      DEFINE FILE 2(130,320,U,K)
      DEFINE FILE 3(40,256,U,K)
      DEFINE FILE 4(130,320,U,K)
      DATA FNUM/1,4/,JD1(4162),JD2(4162)/2*4160/
      DATA DIGIT/440,420,400/,LEV1/50/,LEV2/100/
      DATA M1,M2,M3,M4/6,2,2,1/
      DATA MPT/129*0/
      CALL ZINIT(FNUM,2,IFG)
      READ (2'1) BF1
      CALL ZWRIT(2,2,JD1(4162))
      READ (8,100) ECF,NLG,NFRM,IFG,MT,ITHRS,ISTOP
      WRITE (1,200) EOF,NLG,NFRM,IFG,MT,ITHRS,ISTOP
      IF (MT) 396,396,395
  395 CALL XPOSF(M1,M2,M3,M4)
  396 ECF=ECF-4*NLG+4
      CALL FMODE(M3,M4)
      READ (3,100) NGX
      WRITE (1,200) NGX
      READ (3,100) GL2
      READ (3,100) IGL2
```

```
      IF (IFG) 411,411,413
  413 WRITE (1,333)
      WRITE (1,200) GL2
      WRITE (1,250) IGL2
  411 CONTINUE
      CALL ZTEST(2,1X,1)
      READ (2,14) BF1
      CALL ZWRIT(2,1,JD1(4162))
      DO 585 I=1,4
      READ (8,100) (GN(J,I),J=1,15)
      READ (8,100) (IGN(J,I),J=1,15)
      IF (IFG) 585,585,589
  589 WRITE (1,200) (GN(J,I),J=1,15)
      WRITE (1,250) (IGN(J,I),J=1,15)
      WRITE (1,333)
  585 CONTINUE
      CALL ZTEST(2,IFG,1)
      WRITE (5,489)
      DO 808 IFRM=1,NFRM
      NCFG=0
      NRC=1
      DO 800 IF=1,2
      CALL FATOD(IFF,ECF,1)
      ILN=ECF-4
      DO 800 ILG=1,NLG
      ILFX=MOD(IFRM-1,60)
      CALL ZTEST(1,IFG,1)
      DO 1001 IP=1,2
      BF1(2049,IP)=2048
      IF (ILFX) 672,671,672
  672 CALL ZREAD(1,NRC,JD2(4162))
  671 IFG=NLG-ILG-IF+2
      CALL FOALM(0,BF1(2049,IP))
      IF (IFG)  797,797,798
  798 CALL FATOD(IFM,ILN,1)
      ILN=ILN-4
  797 CALL SBFM(BF1(2048,IP),BF1(1536,IP),512,128)
      CALL SHM2(BF1(2048,IP),BF1(1024,IP),1024)
 1001 CONTINUE
      CALL ZTEST(1,IFG,1)
```

```
      DO 1047 IP=1,2
      IF (IDFX) 1005,1006,1005
 1006 CALL MEAN(125,BF1(2048,IP),BF1(1,IP),0)
      CALL CBRGT(GL2(32),32,IGL2(32),BF1(2048,IP),125,BF1(1,IP))
      GOTO 1037
 1005 IRX=2
      IX=0
      IY=2048
      DO 1030 NT=1,128
      IFG=ISS(BF1(IY,IP),BF2(IY,IP))
      IF (IFG-ITHRS) 1031,1032,1032
 1032 MPT(IRX)=IX
      IRX=IRX+1
      IX=4
      GOTO 1030
 1031 IX=IX+4
 1030 IY=IY-4
      IRX=IRX-2
      MPT(1)=IRX
      NCHG=NCHG+IRX
      IF (IRX) 1047,1047,1048
 1048 IDFX=-40
      CALL IDIFF(BF1(2048,IP),BF2(2048,IP),MPT(1))
      CALL IBRGT(GL2(32),32,IGL2(32),MPT(1),BF1(2048,IP))
      CALL IRENM(BF2(2048,IP),BF1(2048,IP),MPT(1))
 1037 DO 557 I=1,15
      IX=NGX(I)
      IFG=4-I/4
      IFG=IFG*512-NGG(1,4)
      IF (IX+IDFX) 1041,1045,1046
 1041 CALL MLRGT(GN(15,IX),GN(1,IX),IGN(15,IX),MPT(1),BF1(IFG,IP),
     CBF2(IFG,IP))
      GOTO 557
 1046 CALL LNGT(GN(15,IX),GN(1,IX),IGN(15,IX),BF1(IFG,IP),128)
      GOTO 557
 1045 CALL SETVL(BF1(IFG,IP),128,0)
  557 CONTINUE
 1047 CONTINUE
      IF (IDFX) 622,621,800
  621 CALL ZWRIT(1,NRC,OD1(4162))
```

```
      GOTO 800
  622 CALL ZWRIT(1,NRC,JD2(4162))
  800 NRC=NRC+1
      WRITE (5,488) IFRM,NCHG
      CALL FMCDF(M1,M2)
      CALL BLANK(1,ECF,2,BOF)
      CALL ZTEST(1,IFG,1)
      CALL ZREAD(1,1,JD1(4162))
      CALL BLANK(2,ECF,1,BOF)
      ILG=2
      ILK=2
      DO 1002 IFM=1,2
      ILN=ECF
      DO 1002 MT=1,NLG
      CALL ZTEST(1,IFG,1)
      IFG=NLG-MT+2-IFM
      IF (IFG) 985,985,166
  166 IF (ILK-1) 984,984,986
  984 CALL ZREAD(1,ILG,JD1(4162))
      GOTO 985
  986 CALL ZREAD(1,ILG,JD2(4162))
  985 DO  757 IF=1,2
      IF (ILK-1) 381,381,382
  382 BF1(2049,IF)=2048.
      CALL SBHM(BF1(2048,IF),BF1(1536,IF),512,128)
      CALL SHM2(BF1(2048,IF),BF1(1024,IF),1024)
      CALL LMRA4(BF1(2048,IF),2048,0,4096)
      CALL WR4LN(0,BF1(2049,IF))
      GOTO 777
  381 BF2(2049,IF)=2048
      CALL SBHM(BF2(2048,IP),BF2(1536,IP),512,128)
      CALL SHM2(BF2(2048,IP),BF2(1024,IP),1024)
      CALL LMRA4(BF2(2048,IP),2048,0,4096)
      CALL WR4LN(0,BF2(2049,IF))
  777 CALL FDTCA(IFM,ILN,1)
  757 ILK=ILN-4
      ILG=ILG+1
      ILK=MOD(ILK,2)+1
 1002 CONTINUE
      CALL ZTEST(1,IFG,1)
```

```
      DO 681 IX=1,2
      DO 681 IGP=1,2048
  681 BF1(IGP,IX)=LEV2
      CALL WR4LN(0,BF1(2049,1))
      IF=ILN
      MI=ILN-12
      DO 466 IFM=1,2
      GOTO (771,772),IFM
  771 CALL ZTEST(1,IFG,1)
      CALL ZREAD(2,1,JU1(4162))
      GOTO 773
  772 CALL ZTEST(2,IFG,1)
      CALL ZREAD(2,2,JU2(4162))
  773 CONTINUE
      CALL FOTOA(IFM,MI-8,0)
      CALL FOTOA(IFM,MI-4,0)
      CALL FOTOA(IFM,MI,0)
  466 CALL FOTOA(IFM,ILN,1)
      I=IFRM
      CALL ZTEST(2,IFG,1)
      DO 682 ILG=1,3
      IY=MOD(I,10)+1
      I=I/10
      READ (3*IY) CHAR
      IHX=1
      DO 864 ILN=1,4
      DO 864 IFM=1,2
      IX=512*(5-ILN)-DIGIT(ILG)
      DO 865 MI=1,16
      IFG=CHAR(MI,IHX)
      CALL SGN(IFG)
      BF2(IX,IFM)=IFG*LEV2+LEV1
      IFG=CHAR(MI,IHX+8)
      CALL SGN(IFG)
      BF1(IX,IFM)=IFG*LEV2+LEV1
  865 IX=IX-1
  864 IHX=IHX+1
  682 CONTINUE
      ILN=IF-4
      DO 691 IFM=1,2
```

```
      BF1(2049,IFM)=2048
      BF2(2049,IFM)=2048
      CALL WR4LN(0,BF1(2049,IFM))
      CALL FDTOA(IFM,ILN,0)
      CALL WR4LN(0,BF2(2049,IFM))
  691 CALL FDTOA(IFM,ILN-4,0)
      IF (ISTOP) 799,799,788
  788 CALL FMODE(M4,M3)
      READ (6,100) ISTOP
  799 CALL XPOSE(M1,M2,M3,M4)
      M1=M3-1
      CALL MOVE(0,1,MT)
      CALL MOVE(1,1,MT)
  808 CONTINUE
      CALL MOVE(0,0,2)
      WRITE(1,444)
      CALL EXIT
  100 FORMAT(815)
  200 FORMAT (5X,8I10)
  250 FORMAT(8I10)
  333 FORMAT ('****')
  444 FORMAT('*END PROCESSING*')
  488 FORMAT(I4,10X,2I10)
  489 FORMAT ('FRAME',5X,'NUMBER OF CHANGES')
      END
```

VARIABLE ALLOCATIONS

```
   JD1(I )=105D-001C   BF1(I )=105D-005C   JD2(I )=209F-105E   BF2(I )=209F-109E   CL2(
    CH(I )=211D-20E2   IBN(I )=2159-211E   NCX(I )=2168-215A  CHAR(I )=2268-2169 DIGIT(
   ECF(I )=226E        EOF(I )=226F        MFT(I )=22F0-2270     F(I )=22F1         IFG(
  NERR(I )=22F4         MT(I )=22F5      ITHRS(I )=22F6      ISTOP(I )=22F7         M1(
    M3(I )=22FA         M4(I )=22FB        IX(I )=22FC         J(I )=22FD          J(
  NCHG(I )=2300        NRC(I )=2301       IFN(I )=2302        ILN(I )=2303        ILG(.
    IF(I )=2306        IRX(I )=2307       IY(I )=2308        IDK(I )=2309        IGF(.
  LEV1(I )=230C
```

STATEMENT ALLOCATIONS

```
 100  =232A  200  =2320  250  =2331  333  =2334  444  =2338  488  =2342  489  =2347  49
 411  =23F4  589  =2443  585  =2479  672  =24CC  671  =24D8  798  =24F4  797  =24FF  10
1032  =25D6 1031  =25EB 1030  =25F1 1048  =2616 1037  =2608 1041  =26BC 1046  =270D 10
 621  =2771  622  =277F  800  =278B  166  =27F3  984  =27F9  986  =2807  985  =2813  98
```

Final Report [1], otherwise, they are described as follows:

1) XPOSE (M1, M2, M3, M4):

exchange M1 and M2, and

exchange M3 and M4

Let A, B, Q, IQ and M be arrays, and let N1, N2, N3, N4, N5, N6 and M(N4) be positive integers.

2) IDIFF, IRENEW (A(N1), B(N2), M(N4)):

These operations replace the set of numbers:

A(N1), A(N1+M(j))

by the differences (for IDIFF), or the sums (for IRENEW):

$A(N1)\pm B(N2)$, $A(N1+M(j)) \pm B(N2+M(j))$,

for j=N4+1, N4+2, ..., N4+M(N4).

3) IBNQT, or ILNQT(Q(N1), N2, IQ(N3), M(N4), A(N5))

These operations replace the set of numbers:

A(N5), A(N5+M(j)) for j=N4+1, ..., N4+M(N4)

by their inverse quantized values. The quantization cutpoint table is stored in the array

$Q(N1) \geq Q(N1+1) \geq \ldots, Q(N1+N2-2)$,

and the inverse look up table is stored in the array:

IQ(N3), IQ(N3+1), ..., IQ(N3+N2-1).

The number of quantization levels is N2. IBNQT uses a binary search method, while ILNQT uses a linear search method.

4) MBNQT, or MLNQT (Q(N1), N2, IQ(N3), M(N4), A(N5), B(N6))

These operations are almost identical to (3), except the inverse quantized values are put in the array:

B(N6), B(N6+M(j)), j=N4+1,..., N4+M(N4),

(i.e., the old data are not destroyed).

5) ISS (M1, M2) = |M1-M2|,

for any integers M1 and M2.

5.1 Video Tape Presentation

The results of video simulations on the motion detection algorithm are recorded on a video tape which is accompanied with this report. Various input video sequences of different types of motions are experimented. Each sequence is identified by a sample reference numbered which is indicated at the beginning of each sequence and at the upper left-hand corner of each frame. (The number indicated at the upper right-hand corner of each frame is the relative frame number of each sequence.) The percentages of motion regions detected and the corresponding source rates achievable without abridging are shown in Table 5.1.

The video tape was recorded by a Sonny video tape recorder, model AV3560, which, unfortunately, contributes a significant amount of recording noise and synchronization instability. The synchronization instability is particularly noticeable and objectionable at the beginning of each recording sequence. To enhance observation detail, each sequence is instant replayed at a slow motion rate of 1:5.

It is worthwhile to emphasize that a substantial portion of motion regions detected is caused by the simulation noises, such as the signal variations between heads, long term hardware stability drift, etc., which are mentioned in the previous section. In a well implemented realtime system, their effects are negligible; hence, a much improved video quality and efficient source data rate can be expected.

Sample Reference	Percentage of Motion Regions Detected			Achievable source rate without sequence abridging * **
	Min.	Max.	Average	
I	11%	28%	15%	≃ .62 bit/pel
II	8%	20%	12%	≃ .46 bit/pel
III	10%	25%	17%	≃ .56 bit/pel

* assuming 1/16 bit/pel is used to implement the synchronization codewords.

** much lower source rate can be achieved with some sequence abridging

Table 5.1 Percentage of Motion Regions Detected

REFERENCES

1. "Study of Efficient Video Compression Algorithm for Space Shuttle Applications," Final Report, LINKABIT Corporation, San Diego, CA.

2. Landau, H. J., and Slepian, D., "Some Computer Experiments in Picture Processing for Bandwidth Redunctions," The Bell System Technical Journal, Volume 50., No. 5, May-June, 1971.

3. Candy, J.C., Frank, M.A., Haskell, B.G., and Mounts, F.W., "Transmitting Television as Clusters of Frame-to-Frame Differences," The Bell System Technical Journal, Volume 50, No. 6, 1971.

4. Enomoto, H., and Shibata, H., "Orthogonal Transform Coding System for Television Signals," Symposium of Applications of Walsh Functions," 1971, Washington, D.C.

5. Cornsweet, T.N., Visual Perception, Academic Press, 1970.

6. LIM Video Controller Operations Manual, LINKABIT Corporation, San Diego, CA

7. Payne, S.E., The Construction of Hadamard Matrics, Technology Incorporated, Dayton, Ohio, Publication No. ARL 73-0117, July, 1973.

8. Fink, D.G., Television Engineering Handbook, McGraw-Hill Book Co., 1957, New York.

9. Viterbi, A.J., "Convolutional Codes and Their Performance in Communication Systems", IEEE, Vol. COM-19, No. 15, Oct. 1971.